



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROZŠÍŘENÁ REALITA PRO DESKOVÉ HRY

AUGMENTED REALITY FOR DESK GAMES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JURAJ BAČOVČIN

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÍTĚZSLAV BERAN, Ph.D.

BRNO 2018

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2017/2018

Zadání bakalářské práce

Řešitel: **Báčovčín Juraj**

Obor: Informační technologie

Téma: **Rozšířená realita pro deskové hry**
Augmented Reality for Desk Games

Kategorie: Uživatelská rozhraní

Pokyny:

1. Prostudujte problematiku tvorby grafických uživatelských rozhraní a testování uživatelské zkušenosti. Seznamte se s problematikou rozšířené reality, tangible interfaces a interaktivním zařízením ARTable.
2. Vyberte vhodnou deskovou hru a navrhnete její realizaci pro ARTable. Zaměřte se na různou úroveň využití systému při hře.
3. Vyberte vhodné nástroje a navržený systém implementujte. Vstupy systému simulujte nebo využijte externí řešení.
4. Řešení demonstруйте na vybrané stolní zábavné nebo vzdělávací hře.
5. Proveďte testování na uživateli a vyhodnoťte uživatelskou spokojenost.
6. Vytvořte plakát a krátké video prezentující klíčové výsledky vašeho řešení.

Literatura:

- B. Shneiderman, C. Plaisant. *Designing the user interface: strategies for effective human-computer interaction*, Addison-Wesley Computing, ISBN-10: 0-321-53735-1, 2009
- L. Mathis. *Designed for Use: Create Usable Interfaces for Applications and the Web*, Pragmatic Bookshelf, ISBN-10: 1934356751, 2011
- L. Shklar, R. Rosen. *Web Application Architecture: Principles, Protocols and Practices*, Wiley, ISBN-10: 047051860X, 2009

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Beran Vítězslav, Ing., Ph.D., UPGM FIT VUT**

Datum zadání: 1. listopadu 2017

Datum odevzdání: 16. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
L.S. 602 66 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Táto bakalárska práca sa zaoberá realizáciou vybranej spoločenskej hry pre interaktívny stôl. Jej cieľom je navrhnuť, implementovať a otestovať aplikáciu s rozhraním, ktorého účelom je dosiahnutie čo najpozitívnejšej používateľskej skúsenosti bez straty kľúčových aspektov definujúcich hranie stolových hier. Výsledný program je spustiteľný v pracovnom priestore ARTable3, umožňujúcom premietanie obrazu na dotykovú plochu a snímanie udalostí z jej povrchu v reálnom čase. Bol podrobený riadnemu funkcionálnemu a používateľskému testovaniu, takže ho je možné považovať za plne funkčný a pripravený na prípadné rozšírenia navrhované v záverečných častiach tejto práce.

Abstract

This bachelor thesis deals with the realization of a chosen board game for an interactive table. The goal is to design, implement and test a new user-friendly user interface of the game application without losing any key aspects defining board gaming. The resulting program is executable on the ARTable3 workspace capable of projecting any image onto the touch table and capturing real-time events from its surface. It has undergone proper functional and usability testing, making it possible to consider it fully functional and ready for any expansions proposed in the final sections of this document.

Kľúčové slová

Záchranári, rozšírená realita, stolové hry, spoločenské hry, grafické používateľské rozhranie, hmotné používateľské rozhranie, ARTable, interaktívny stôl, používateľské testovanie, kalibrácia

Keywords

Rescuers, augmented reality, desk games, board games, graphical user interface, tangible user interface, ARTable, interactive table, usability testing, calibration

Citácia

BÁČOVČIN, Juraj. *Rozšírená realita pro deskové hry*. Brno, 2018. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Vítězslav Beran, Ph.D.

Rozšířená realita pro deskové hry

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Vítězslava Berana, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Juraj Bačovčín
16. mája 2018

Podakovanie

Touto cestou by som chcel poďakovať vedúcemu mojej bakalárskej práce, Ing. Vítězslavovi Beranovi, Ph.D., za opakované nasmerovanie správnou cestou nielen v oblasti samotného návrhu, ale aj výsledného písomného dokumentu. Zároveň by som však chcel poďakovať i ďalším členom výskumnej skupiny robotiky Robo@FIT za cenné rady ohľadom práce s interaktívnym stolom a všetkým účastníkom používateľského testovania prototypu aplikácie.

Obsah

1	Úvod	2
2	Kľúčové znalosti	3
2.1	Používateľské rozhrania	3
2.2	Interaktívny stôl	8
2.3	Testovanie softvéru	10
2.4	Stolové hry	12
3	Analýza a návrh riešenia	14
3.1	Analýza požiadaviek	14
3.2	Herné stavy a funkcionality	16
3.3	Návrhové modely	19
4	Popis implementácie	22
4.1	Implementačné nástroje	22
4.2	Úvodná obrazovka s nastaveniami	23
4.3	Herná obrazovka	24
4.4	Integrácia do robotického operačného systému	27
5	Testovanie aplikácie	30
5.1	Funkcionálne testovanie	30
5.2	Plán používateľského testovania	30
5.3	Výsledky používateľského testovania	33
6	Záver	36
	Literatúra	37
A	Obsah CD	39
B	Dotazník	40
C	Odpovede na dotazník	41
D	Používateľský manuál	42

Kapitola 1

Úvod

Hry sú odjakživa nepochybne neoddeliteľnou súčasťou života väčšiny ľudí, či už sa jedná o športové, mobilné, konzolové alebo akékoľvek iné. Niet preto najmenšieho divu, že vo veľkej miere definujú tvár zábavného priemyslu, konkurujú dokonca i dnes všetkým ostatným formám trávenia voľného času. Iba málokto si však uvedomuje, že ak vezmeme do úvahy celkový dopad na našu civilizáciu, medzi tie najdôležitejšie patria stolové. Boli vonkoncom doby, kedy napríklad šach ovplyvňoval aj niektoré skutočné vojny, no hoci tie už pominuli a príchod nových technológií zdanlivo vytlačil spoločenské hry do ústrania predovšetkým podnikovej zábavy, ich nezanedbateľné čaro im umožnilo pretrvať i v rodinných kruhoch.

Cieľom mojej bakalárskej práce je realizovať konkrétnu stolovú hru s využitím modernej techniky a umožniť jej doplnenie atraktívnymi multimediálnymi prvkami bez straty kľúčových pozitívnych aspektov definujúcich jej hranie za účelom poskytnutia čo najlepšej používateľskej skúsenosti. Dosiahnuť ho pomáha najmä technológia rozpoznávania podnetov z fyzického sveta vedúceho k projekcii scény videnej používateľmi, rozšírená realita.

Na ilustráciu konkrétneho postupu uplatnenia daného konceptu bola predovšetkým v snahe vyhnúť sa štandardným voľbám vybraná kooperatívna hra Záchranári od vydavateľstva Mindok. Tento dokument detailne pojednáva o jej prevedení do aplikácie spustiteľnej s využitím interaktívneho stola, vyznačujúceho sa okrem iného práve schopnosťou snímať dotyky z jeho povrchu a príslušne na ne reagovať premietaním obrazu na hraciu plochu.

Z hľadiska štruktúry je moja práca rozčlenená do nasledujúcich logických celkov v zachovanom poradí: najdôležitejšie poznatky potrebné k dosiahnutiu vyššie definovaného cieľa vrátane bližšieho popisu spoločenskej hry Záchranári, návrh jej zmodernizovanej podoby pre interaktívny stôl, implementačné detaily a prevedené testovanie aplikácie s možnými rozšíreniami vyplývajúcimi z jeho výsledkov. Záver obsahuje už iba stručnú sumarizáciu všetkých jej najkľúčovejších výstupov a námetov na potenciálne budúce vylepšenia programu.

Kapitola 2

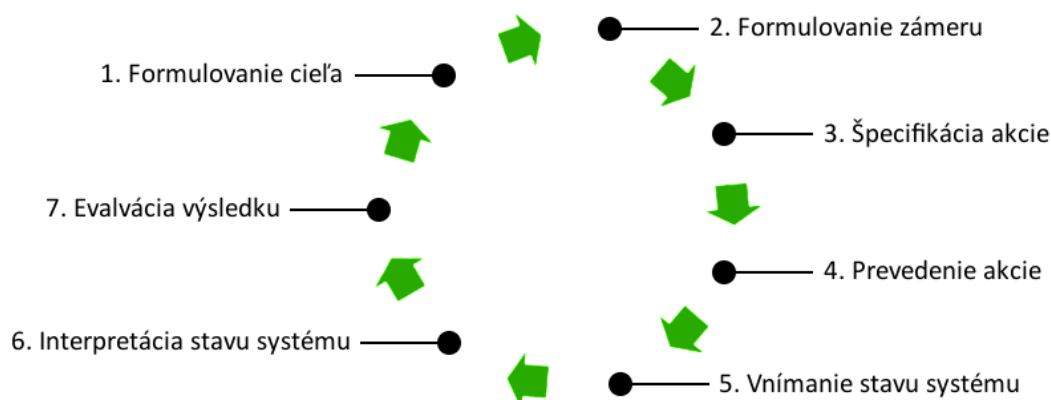
Kľúčové znalosti

Táto kapitola poskytuje čitateľom všetky najdôležitejšie informácie potrebné na realizáciu vybranej stolovej hry v rozšírenej realite. Zaoberá sa predovšetkým teóriou z oblasti používateľských rozhraní, bližšie popisuje interaktívny stôl, vysvetľuje základné princípy testovania softvéru a definuje najvýznamnejšie aspekty spoločenského hrania. Stručne tak tiež i predstavuje zvolenú hru Záchranári od vydavateľstva Mindok, ktorej modernizácia a možnosť rozšírenia pridaním multimediálnych prvkov je jadrom tejto bakalárskej práce.

2.1 Používateľské rozhrania

Používateľské rozhranie (User Interface, UI) je v oblasti interakcie človeka s počítačom možné definovať ako priestor, v ktorom dochádza k vzájomnému pôsobeniu medzi ľuďmi a strojmi [18]. Ako to už vyplýva i z názvu, jeho primárnou úlohou je umožnenie a uľahčenie používania zariadenia alebo programu zadávaním vstupov za účelom dosiahnutia požadovaných výstupov. Typicky je tento spôsob komunikácie vhodné interpretovať ako cyklus nasledujúcich siedmich etáp činností zo strany používateľa (pozri obrázok 2.1):

1. **Formulovanie cieľa** - definovanie výsledku, ktorý by používateľ chcel dosiahnuť (napríklad snaha o nejakú konkrétnu prácu v ľubovoľnom programe)
2. **Formulovanie zámeru** - definovanie plánu vedúceho k vyššie popísanému cieľu (napríklad spustenie daného programu)
3. **Špecifikácia akcie** - adaptovanie spomínaného zámeru do podoby konkrétnej akcie (napríklad nutnosť stlačenia tlačidla pre spustenie programu)
4. **Prevedenie akcie** - realizácia špecifikovanej akcie používateľom (napríklad stlačenie tlačidla pre spustenie daného programu)
5. **Vnímanie stavu systému** - uistenie sa, že zariadenie pracuje, na základe nejakej spätnej väzby (napríklad spúšťanie požadovaného programu)
6. **Interpretácia stavu systému** - porozumenie tomu, k čomu viedla špecifikovaná akcia (napríklad otvorenie nejakého programu)
7. **Evalvácia výsledku** - zhodnotenie, či v dôsledku prevedenej akcie došlo na požadovanú reakciu (napríklad otvorenie správneho programu)

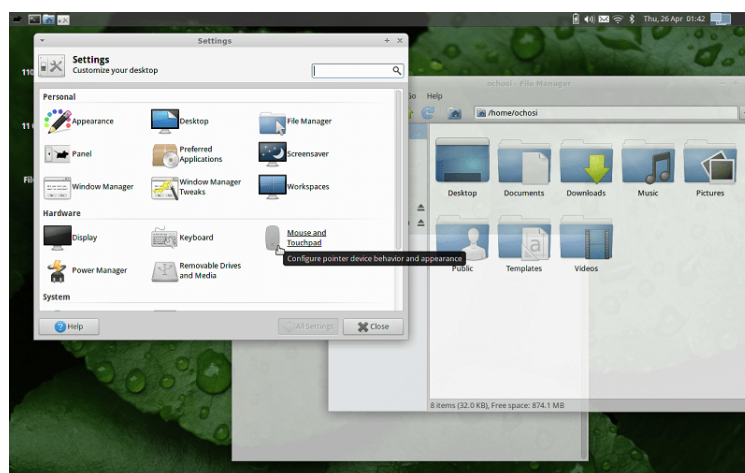


Obr. 2.1: Sedem etap interakcie používateľa so zariadením podľa [8]

Používateľské rozhrania najčastejšie delíme do skupín na základe rozlišnej reprezentácie zadávaných vstupov a zobrazovaných výstupov. Typickým príkladom sú **rozhrania príkazového riadka (Command-Line Interface, CLI)**, prostredníctvom ktorých človek komunikuje so zariadením alebo programom s využitím textu. Pre potreby tejto práce sa však je nutné zamerať iba na **grafické používateľské rozhrania (Graphical User Interface, GUI)** a **hmotné používateľské rozhrania (Tangible User Interface, TUI)**.

Grafické používateľské rozhrania

Rozhranie reprezentujúce vstupy a výstupy pomocou grafických prvkov alebo iných vizuálnych indikátorov sa nazýva **grafické používateľské rozhranie (Graphical User Interface, GUI)** [14]. Jeho hlavným cieľom je umožniť ľuďom zaobchádzať so zariadeniami intuitívnym spôsobom bez nutnosti pamätania si veľkého počtu rozličných textových príkazov. Jedná sa preto v porovnaní s rozhraním príkazového riadka v súčasnosti už po dobu niekoľkých rokov o podstatne rozšírenejšiu formu interakcie človeka so strojom¹.



Obr. 2.2: Ukážka typickej podoby GUI prevzatá z [18]

¹Command line vs. GUI, <https://www.computerhope.com/issues/ch000619.htm>

Grafické používateľské rozhrania obyčajne pozostávajú z okien a plôch obsahujúcich elementy, ktoré je možné rozdeliť do štyroch nasledujúcich zaužívaných skupín [18]:

- **vstupné prvky** - textové pole, posúvač (slider), adresný riadok, atď.
- **výstupné prvky** - textový popis, stavový riadok, indikátor priebehu, atď.
- **výberové prvky** - výberový zoznam (listbox), rozbaľovací zoznam (drop-down list), kombinované pole (combobox), zaškrŕavacie pole (checkbox), atď.
- **akčné prvky** - predovšetkým rôzne typy tlačidiel s rozmanitým dizajnom

Súčasťou návrhu takmer akejkoľvek modernej aplikácie je práve rozvrhnutie a umiestnenie predovšetkým mnohých z vyššie vymenovaných elementov, väčšinou prezentovaných na nejakej obrazovke a ovládateľných myšou, klávesnicou alebo dotykom. Nie je tomu teda inak ani v prípade programu, ktorý je cieľom tejto bakalárskej práce, no na rozdiel od tradičných grafických používateľských rozhraní využíva i niektoré koncepty predstavené nižšie.

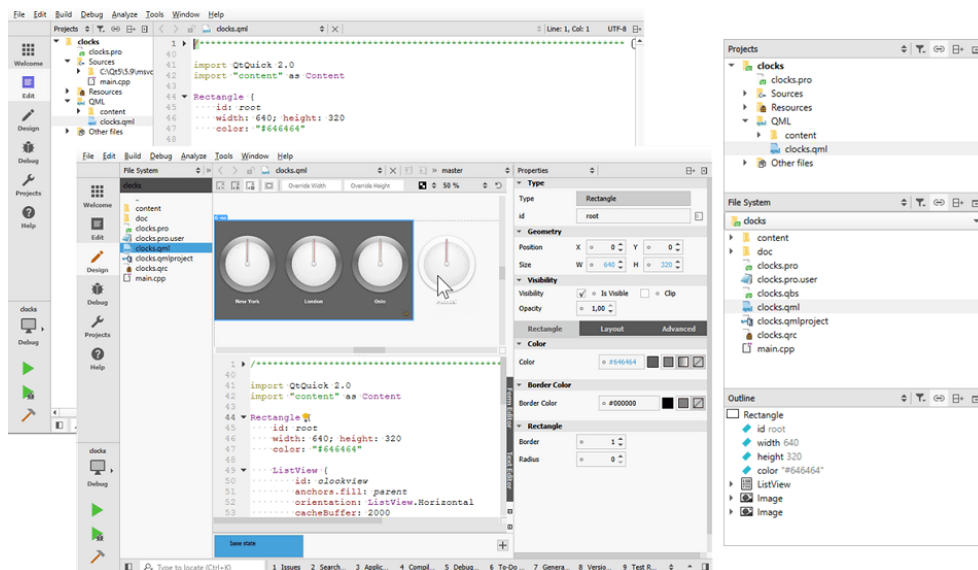
Nástroje pre tvorbu grafických používateľských rozhraní

K realizácii grafických používateľských rozhraní existuje veľmi veľa rozličných prístupov, líšiacich sa predovšetkým na základe využitého programovacieho jazyka, cieľovej platformy a požiadaviek na vyvíjanú aplikáciu. S pribúdajúcimi nárokmi najmä na jednoduchosť, efektívnosť a prenositeľnosť za posledných pár desiatok rokov došlo k pozoruhodnému rozšíreniu softvérových nástrojov pre tvorbu GUI, obyčajne umožňujúcich i priebežné vizuálne úpravy, čo vo výsledku poskytuje veľkú kontrolu nad celkovým dizajnom [13]. Pre potreby tejto bakalárskej práce sa však nie je nevyhnutné zaoberať napríklad takými **mobilnými** (AIDL - Android Interface Definition Language) a **webovými** (HTML, CSS, PHP, atď.) technológiami, pretože jej výstupom z konceptuálneho hľadiska je **desktopová** aplikácia.

Medzi najpopulárnejšie multiplatformné *GUI toolkity* (sady nástrojov na tvorbu grafických používateľských rozhraní), vhodné pre vývoj v desktopovom prostredí, patria **ILOG Views** (založené na C++), **GTK+** (C, C++, Perl, Ruby, Java, atď.) a **Qt** (C++, C#, Pascal, Haskell, Python, atď.), z ktorých posledné dva sú dokonca dostupné aj zdarma [13]. Predovšetkým vďaka jednoduchej integrácii projektov do **robotického operačného systému** (pozri nižšie) a predchádzajúcim skúsenostiam bol pre túto prácu zvolený aplikačný rámec **Qt 5**, ktorý okrem iného taktiež zdokonaľuje prácu s deklaratívnym jazykom **QML**.

Qt využíva na popis grafického používateľského rozhrania stromovú hierarchiu s umiestnením elementov daným rozvrhnutím *layout* a chovaním určeným ich zdrojovým kódom so získanými dedičnosťami, ktorej základným objektom je ovládací prvok *widget*. Poskytuje niekoľko vlastných nástrojov (pozri obrázok 2.3) vrátane prekladového (qmake), dizajnového (Qt Designer), lokalizačného (Qt Linguist) a dokumentačného (Qt Assistant), pričom prostredníctvom modulu *QT QML* predstavuje i zmieňované **QML**. Jedná sa o programovací jazyk s objektovým zápisom na báze notácie *JSON* s podporou imperatívnych výrazov *JavaScriptu*, ktorý umožňuje popisom prepojených vizuálnych komponentov, ich vzájomných interakcií a vzťahov vytvárať vysoko výkonné, plynulo animované, atraktívne aplikácie [16]. Jeho základným stavebným kameňom je objekt *Item* so všetkými hlavnými vizuálnymi vlastnosťami (šírka, výška, ukotvenie, atď.), ktorý je zároveň v hierarchii predkom ostatných elementov. Štandardnou knižnicou obsahujúcou typy a funkcionality **QML** je *Qt Quick*. Bližšie súvisiace informácie je možné získať z **dokumentácie ku QT** [16].

²Qt Creator, <http://doc.qt.io/qtcreator/>



Obr. 2.3: Ukážka vývojového prostredia Qt, Qt Creator²

Hmotné používateľské rozhrania

V posledných pár desiatkach rokov sa na priemyselnú scénu dostávajú rozhrania poskytujúce používateľovi hmatateľnú reprezentáciu digitálnych informácií. Prináleží im názov **hmotné používateľské rozhrania (Tangible User Interfaces, TUI)**, pričom ich primárna funkcia spočíva v prepojení rôznych fyzických objektov s dátami za účelom obohatenia práce so vstupmi a výstupmi vedúceho k paralelným cyklom spätných väzieb (haptická spätná väzba informujúca o prevedení akcie a jej audiovizuálna interpretácia v systéme) [12].



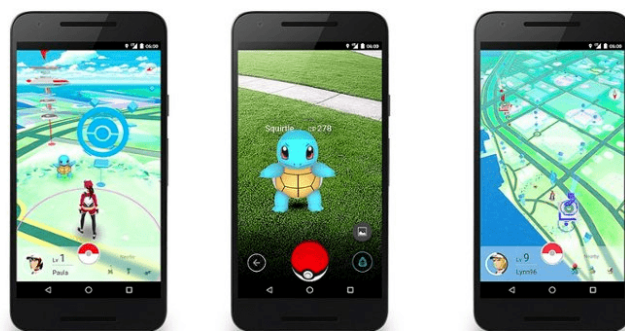
Obr. 2.4: Ukážka konkrétneho TUI, Reactable³, prevzatá z [18]

³Reactable - Music Knowledge Technology, <http://reactable.com/>

V súčasnosti už predovšetkým vďaka ich atraktivnosti a precíznosti ovládania existuje pomerne veľa rôznych podôb a uplatnení TUI, siahajúc od zábavného priemyslu cez geofyziku až po molekulárnu biológiu [12]. V kontexte tohto dokumentu je však tým najzaujímavejším prevedením takzvaná **hmotná rozšírená realita (Tangible Augmented Reality, TAR)**, kombinujúca hmatateľný vstup s vizuálnym výstupom rozšírenej reality.

Rozšírená realita

V obecnom ponímaní je **rozšírená realita (Augmented Reality, AR)** technológiou rozpoznávania podnetov z fyzického sveta vedúceho k projekcii obrazu prekrývajúceho a rozširujúceho scénu vnímanú používateľmi o digitálne informácie v reálnom čase [4]. V oblasti zábavného priemyslu je typickým príkladom dosť známa aplikácia *Pokémon GO*⁴, ktorá využíva skutočnú mapu na rozmiestnenie virtuálnych herných prvkov (videných len prostredníctvom kamery mobilných zariadení) do existujúcich lokalít (pozri obrázok 2.5).



Obr. 2.5: Ukážka konkrétnej AR aplikácie, *Pokémon GO*⁴, prevzatá z NewGenApps⁵

Bližšie k zameraniu bakalárskej práce má však konkrétne **priestorová rozšírená realita (Spatial Augmented Reality, SAR)**, vyznačujúca sa oddelením zobrazovacej technológie od človeka (odňatím z rúk, očí, hlavy, atď.) a jej integráciou priamo do prostredia [2]. Uplatnenie práve tejto techniky je ideálne na dokonalé napodobenie hrania stolových hier s možnosťou zvýšenia jej dynamiky odobraním povinnosti používateľa manuálne počítať dopady pokročilých mechaník a pridania mnohých atraktívnych audiovizuálnych elementov.

O zachovanie a prípadné zlepšenie interaktivity ovládania sa je schopný postarať vyššie spomínaný koncept **hmotnej rozšírenej reality (Tangible Augmented Reality, TAR)**. Tá totiž umožňuje priradiť vstupné digitálne informácie fyzickým objektom (napríklad hracím figúrkam alebo žetónom), ktoré sú zvyčajne súčasťou výstupnej scény SAR (napríklad herná mapa) [12]. Typickým príkladom tomuto podobného, no o čosi zložitejšieho systému, je rozhranie s názvom *Tangible Tiles*⁶, ktoré využíva opticky sledované kachličky z plexiskla pre interakciu a projekciu obrazu na stôl (pozri obrázok 2.6).

⁴Pokémon GO, <https://pokemongolive.com/en/>

⁵New Generation Applications - Cloud Computing, <https://www.newgenapps.com/>

⁶Manuela Waldner - Tangible Tiles, <http://manuela-waldner.net/project.php?id=1>

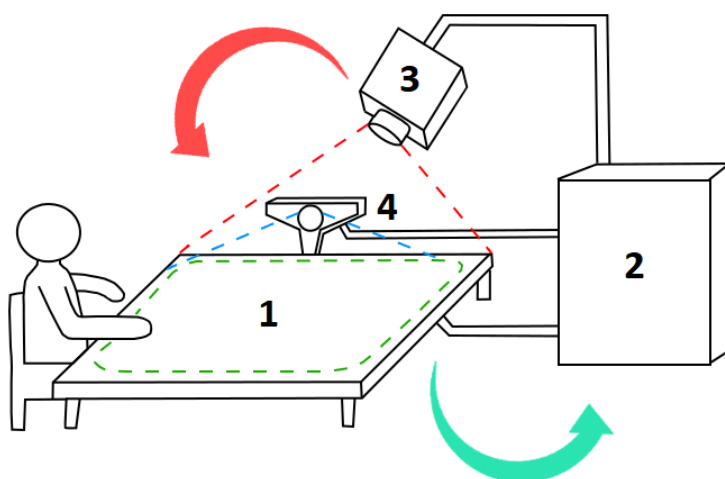


Obr. 2.6: Ukážka konkrétneho rozhrania TAR s názvom Tangible Tiles⁶

2.2 Interaktívny stôl

Základným kameňom interakcie mnohých hmotných rozhraní býva často objekt s mechanizmom schopným snímania rôznych udalostí z jeho povrchu [12]. V kombinácii s projekčnými technikami priestorovej rozšírenej reality a vhodne definovaného chovania pri zmene hernej scény vytvára systém pre jednoduchosť nazývaný ako **interaktívny stôl**, ktorý pozostáva z nasledujúcich najdôležitejších súčastí (pre lepšiu orientáciu pozri obrázok 2.7):

1. vyvýšený povrch, cez ktorý je možné zadávať vstupy aplikácii spustenej zvyčajne na prepojení počítači (napríklad dotyková plocha)
2. riadiaci stroj, ktorý prostredníctvom vybranej aplikácie udržiava vzájomnú komunikáciu medzi všetkými súčastami interaktívneho stola (zvyčajne počítač)
3. zariadenie schopné premietiť výstup na spomínaný povrch (zväčša projektor)
4. detektor pozície, veľkosti, farieb a iných požadovaných vlastností vybraných objektov nachádzajúcich sa v oblasti interaktívneho stola (napríklad hĺbková kamera)



Obr. 2.7: Nákres základných súčastí interaktívneho stola

Presne takýto systém je pod názvom ARTable3 dostupný v robotickom laboratóriu VUT FIT v Brne. Informáciám o jeho hardvérových záležitostiach a spôsobe zostavenia sa preto tento dokument venuje len veľmi okrajovo, pričom riešenie skreslenia obrazu a kalibrácie jednotlivých uzlov interaktívneho stola je popísané až v posledných častiach štvrtej kapitoly.

Pracovný priestor ARTable3

Projekt *ARTable*⁷ v robotickom laboratóriu VUT FIT v Brne vznikol na základe konceptu vymysleného vybranými zástupcami tejto školy za účelom vytvorenia vyčleneného priestoru pre bezpečnú spoluprácu človeka s robotom, využívajúc okrem iného práve rozšírenú realitu s interaktívnym stolom. Väčšina jeho najpodstatnejších komponentov bola dodaná spoločnosťami *Fortes* (projekčná a dotyková kapacitná fólia so špeciálnym vodivým plastom), *Microsoft* (hlbková kamera Kinect) a *BenQ* (dátový projektor obrazu).

V rovnakej miestnosti neskôr vzhľadom na veľký záujem študentov došlo i k zriadeniu podobného pracoviska s názvom **ARTable3** bez spomínaného robota. Priestor teda pozostáva akurát z vysoko responzívnej dotykovkej plochy (100x60 centimetrov) s USB pripojením, statívu na pripevnenie jeho komponentov, projektoru s HDMI vstupom, hlbkovej kamery Kinect a počítača so všetkými nainštalovanými súčastami, ktoré sú nevyhnutné pre riadenie vzájomnej komunikácie medzi uzlami interaktívneho stola (pozri obrázok 2.8).



Obr. 2.8: Pracovný priestor ARTable3 v robotickom laboratóriu VUT FIT v Brne

Pre potreby tejto bakalárskej práce bude postačujúci len dedikovaný počítač, zmieňovaná dotyková plocha a spomínaný projektor. Využitie aplikačné rozhranie interaktívneho stola je obsiahnuté vo verejne dostupnom repozitári **RoboFIT**⁸. Správne spustenie systému zaobstará balík s názvom *art_bringup*, ktorý pripraví aj projekčný uzol premietajúci obraz posielať z grafického používateľského rozhrania zvoleného programu (za predpokladu, že je projektor dobre skalibrovaný). Na čítanie dát z povrchu stola a publikovanie správ so súradnicami stlačených miest, ku ktorým musí mať prístup taktiež samotné GUI (vyžaduje sa kalibrácia dotykovkej plochy), je využívaný balík nazývaný *art_touch_driver*.

⁷FORTES - ARTable na VUT FIT, <http://www.fortes.cz/portfolio/artable-na-vut-fit-2/>

⁸GitHub repozitár, <https://github.com/robofit/artable>

Robotický operačný systém

Pre prácu s interaktívnym stolom ARTable3 je potrebné rozumieť základom týkajúcim sa **robotického operačného systému (Robot Operating System, ROS)**. Jedná sa o open-source softvér poskytujúci služby typického operačného systému (abstrakcia hardvéru, implementácia bežne používaných funkcií, prenos správ medzi procesmi, rôzne vizualizácie, riadenie balíkov, atď.), ktoré sú primárne určené k ovládaniu robota [10]. Umožňuje ale práve i vzájomnú komunikáciu medzi všetkými komponentmi daného pracovného priestoru, v súvislosti s ktorou sa je nevyhnutné oboznámiť predovšetkým s nasledujúcimi pojmami:

- **uzly (nodes)** - spustiteľné súbory, ktoré prostredníctvom robotického operačného systému posielajú a prijímajú správy od ostatných bežiacich uzlov
- **správy (messages)** - dátový typ robotického operačného systému používaný na vzájomné vymieňanie informácií medzi jednotlivými uzlami
- **témy (topics)** - pomenované komunikačné systémy, prostredníctvom ktorých uzly posielajú a prijímajú správy od ostatných uzlov
- **master** - poskytovateľ menovacích a registračných služieb pre všetky uzly v systéme
- **rosout** - rozhranie pre predávanie výstupných dát (ekvivalent stdout/stderr pre ROS)
- **roscore** - kolekcia prerekvizitných uzlov a programov potrebných pre beh ROS (master, rosout a server pre ukladanie alebo načítanie parametrov jednotlivých uzlov)

Na to, aby nový **uzol (node)** mohol naviazať komunikáciu v systéme ROS, musí byť v prvom rade inicializovaný. V závislosti na tom, či má informácie posilať alebo naopak prijímať, následne potrebuje publikovať alebo odoberať konkrétnu **tému (topic)**. Jednotlivé typy **správ (messages)** využívané aplikáciou, ktorej proces návrhu a vytvorenia je predmetom tejto bakalárskej práce, sú obsiahnuté v balíkoch *std_msgs*⁹ (*Bool*, *Empty*) a *art_msgs*¹⁰ (*Touch*). V tom druhom je taktiež možné nájsť **služby (services)** definované párom správ (žiadosť a odpoveď) pre **vzdialené volanie procedúr (remote procedure call, RPC)**. Jedná sa konkrétne o *TouchCalibrationPoints* a *TouchCalibrationPointsResponse*, uplatnené pri kalibrácii dotykovej plochy. Poslednými nevyhnutnými službami sú *Trigger* (využívaný naopak pri kalibrácii projektoru) a *Empty* z balíka *std_srvs*¹¹. Ďalšie súvisiace informácie sú dostupné v dokumentácii robotického operačného systému [10].

2.3 Testovanie softvéru

Či už vo väčšej alebo menšej miere, neoddeliteľnou súčasťou vývoja každej aplikácie je nepochybne empirický technický výskum jej kvality. Nie je tomu samozrejme inak ani pre túto bakalársku prácu, preto sa táto jej časť stručne zaoberá tými najdôležitejšími súvisiacimi pojmami a rozdeleniami, počínajúc vysvetlením samotného **testovania softvéru**.

⁹Balík *std_msgs*, http://wiki.ros.org/std_msgs

¹⁰Balík *art_msgs*, <https://github.com/robofit/artable-msgs>

¹¹Balík *std_srvs*, http://wiki.ros.org/std_srvs

Jedná sa o proces alebo sériu procesov s cieľom uistiť sa, že aplikácia vykonáva všetko podľa návrhu a nič neúmyselné [9]. Výstupom by mal byť zdokumentovaný zoznam objavených chýb a čo najpresnejšia evalvácia kvality zloženej z niekoľkých vrstiev, ktorým sa hovorí **dimenzie kvality**. Jedná sa konkrétne predovšetkým o nasledujúce aspekty vyvíjanej aplikácie závislé na požiadavkách a očakávaniach používateľov [17]:

- **funkčnosť (functionality)** - správne chovanie všetkých funkcií programu
- **použiteľnosť (usability)** - spôsob, akým je z pohľadu používateľa možné dosiahnuť jednotlivé požadované ciele aplikácie (napríklad jednoduchosť ovládania)
- **spoľahlivosť (reliability)** - konzistencia chovania programu za takmer akýchkoľvek okolností (najmä v súvislosti s preťažením) a detekcia prípadných chybových stavov
- **výkon (performance)** - rýchlosť aplikácie a zvládanie čo najväčšieho počtu udalostí
- **podporovateľnosť (supportability)** - jednoduchosť inštalácie a údržby programu

Kritériá, na ktoré je pri testovaní potrebné brať ohľad, je samozrejme i viac (lokalizovateľnosť, kompatibilita, bezpečnosť, prenositeľnosť, integrovateľnosť, atď.). Pre potreby tejto bakalárskej práce sa však je potrebné sústrediť hlavne na **funkčnosť** a **použiteľnosť**.

Funkcionálne testovanie

Výskumom kvality softvéru, ktorý sa zameriava primárne na **funkčnosť** aplikácie, je v tomto dokumente vzhľadom na časové nároky a obmedzené prostriedky iba **funkcionálne testovanie (functional testing)**. Ide o jeden z hlavných typov **testovania čierna skrinka (black-box testing)**, čo v praxi znamená, že sa pri ňom program považuje za kompletnú entitu, jeho vnútorná štruktúra je absolútne ignorovaná a vstupné dáta sú odvodené čisto len zo špecifikácie požadovaných cieľov hodnotenej aplikácie [9]. Zahŕňa sériu procesov vykonávajúcich validáciu správania systému podľa jednotlivých funkcionalít pomocou širokého rozsahu normálnych a chybových údajov porovnávaním reálnych výstupov s očakávanými.



Obr. 2.9: Ilustračný obrázok testovania čierna skrinka (black-box testing) podľa [17]

Tento typ testovania obyčajne pozostáva z nasledujúcich šiestich krokov¹²:

1. identifikácia funkcionalít, ktoré testovaný softvér nesmie postrádať
2. príprava vstupných dát na základe špecifikácie požiadaviek na aplikáciu
3. nachystanie očakávaných výstupov podľa špecifikácie požiadaviek na program
4. vyskúšanie každej jednej identifikovanej funkcionality testovaného softvéru
5. porovnanie výsledkov prevedenia zmieňovaných funkcií s očakávanými výstupmi
6. dôkladná kontrola, či aplikácia naozaj funguje podľa potrieb používateľov

¹²Podľa článku na adrese https://en.wikipedia.org/wiki/Functional_testing

Používateľské testovanie

Vyhodnotenie a zlepšenie použiteľnosti programu je primárnym cieľom **používateľského testovania (usability testing)**. Zameriava sa predovšetkým na nasledujúce kritériá [1]:

- **účelovosť** - miera umožňovania používateľovi plniť požadované ciele
- **efektívnosť** - jednoduchosť používania za účelom splniť požadované ciele
- **užitočnosť** - miera poskytnutia funkcionality potrebnej pre splnenie cieľov
- **naučiteľnosť** - pochopiteľnosť programu a jeho ovládania
- **zapamätateľnosť** - náročnosť zapamätania si postupov vedúcich k splneniu cieľov
- **bezpečnosť** - v tomto kontexte najmä ochrana pred nechcenými akciami

Existuje veľmi veľa rozdielných variácií používateľského testovania, rozlišujúcich sa predovšetkým z hľadiska metodológie (výber a tréning účastníkov, role, dĺžka, lokalita, počet iterácií, atď.). Všetky ale jednoznačne zdieľajú týchto päť hlavných charakteristík [3]:

1. Primárnym zámerom je zlepšenie výslednej aplikácie.
2. Účastníci musia reprezentovať očakávaných cieľových používateľov.
3. Účastníci musia vykonávať skutočné akcie programu, nie iba odpovedať na otázky.
4. Dôležité akcie a prejavy zo strany účastníkov sú pozorované alebo aj nahrávané.
5. Získané informácie sú dôkladne analyzované, všetky najkritickejšie problémy musia byť diagnostikované a spracované výsledky poslúžia na zdokonalenie programu.

Výberom konkrétnych metód získavania spätnej väzby pozorovaním, ktoré je kľúčom k vyhodnoteniu používateľskej spokojnosti, sa podrobnejšie zaoberá až piata kapitola.

2.4 Stolové hry

Stolová hra je akákoľvek hra, ktorá má fixný súbor pravidiel definujúcich počet, pozíciu a všetky možné akcie vzájomne na seba pôsobiacich objektov hráčov obyčajne reprezentovaných figúrkami na fyzickej hracej ploche [5]. Typickým najznámejším príkladom je deterministická strategická hra s názvom *Šach*, ktorá sa v minulosti dokonca aj preukázateľne do značnej miery podpísala na vývoji ľudskej civilizácie. Hoci však už táto forma zábavy v dnešnej dobe nemá až tak obrovský význam, predovšetkým v kruhoch rodiny, blízkych priateľov a rôznych podnikov pretrváva vo veľkom i naďalej.

Existuje niekoľko spôsobov klasifikácie stolových hier, odvíjajúcich sa často predovšetkým od ich kognitívnych aspektov [5]. Jedná sa konkrétne o hry vojenské, pretekárske, zarovnávacíe, počítacie a mnohé ďalšie. Tým najdôležitejším z pohľadu bakalárskej práce je však delenie na kompetitívne (cieľom každého hráča je zvíťaziť nad ostatnými) a koopeatívne (spoločným cieľom všetkých hráčov je zvíťaziť nad nejakým mechanizmom).

Nech už ale patria do ktorejkoľvek z menovaných skupín, pre takmer všetky exempláre platí, že ich najväčším pozitívom je poskytovanie bohatých sociálnych interakcií medzi zúčastnenými, ktoré napríklad konkurujúce videohry vzhľadom na nutnosť sústrediť sa nie na hráča, ale na obrazovku, väčšinou v takejto miere postrádajú. Kľúčom k úspešnej modernizácii stolových hier bez straty ich čara je preto hlavne zachovanie tejto prednosti.

Záchranári od Mindok

Stolová hra, ktorej realizácia pre interaktívny stôl je predmetom tejto bakalárskej práce, sa nazýva **Záchranári (Flash Point: Fire Rescue)**. Jedná sa o spoločenskú zábavu od vydavateľstva Mindok, určenú primárne dvom až šiestim ľuďom starším ako desať rokov, ktorej cieľom je urýchlene vyniesť čo najväčší počet ohrozených osôb z horiacej budovy [15].

Hráč sa zmocňuje role jedného zo záchranárov a v spolupráci s ostatnými robí všetko preto, aby dospel k požadovanému záveru. V rámci každého kola môže uskutočniť štyri akcie pozostávajúce z **pohybu**, **hasenia**, **otvárania** či **zatvárania** dverí, **búrania** stien a **vynášania** zranených osôb. Môže si ich ale prípadne aj **ušetriť** do ďalšieho kola, aby toho stihol vykonať viac naraz. Nesmie však otáľať prídlho, pretože dym a oheň sa nekompromisne šíri, spôsobujúc čoraz väčšie škody na ľuďoch i majetku, ktoré môžu viesť k prehre v prípade usmrtenia príliš vysokého počtu ohrozených jedincov alebo zrútenia budovy.



Obr. 2.10: Obrázok stolovej hry Záchranári prevzatý z [15]

Kompletný návod k originálnemu zneniu hry a podrobný popis všetkých jej komponentov je k dispozícii v prílohách (jeden zo súborov obsiahnutých na pamäťovom médiu).

Kapitola 3

Analýza a návrh riešenia

Táto kapitola sa zaoberá v prvom rade analýzou a definíciou problémov vyplývajúcich z tej predchádzajúcej, ktorých riešenie je kľúčové pre naplnenie cieľov práce. V ďalšej časti pojednáva o návrhu aplikácie inšpirovanej vybranou stolovou hrou *Záchranári* pre interaktívny stôl, sústrediac sa predovšetkým na stručný popis najdôležitejšej plánovanej funkcionality (vrátane herných akcií a stavov) nasledovaný dátovým a procesným modelovaním.

3.1 Analýza požiadaviek

Niet najmenších pochyb o tom, že kľúčom k efektívnemu návrhu akejkoľvek aplikácie je v prvom rade dôkladná **analýza požiadaviek**, počínajúc samotnou definíciou problémov, ktoré by výsledný program mal riešiť. Nezáleží totiž na tom, do akej miery je používateľské rozhranie prepracované a atraktívne, pokiaľ je samotná funkcionality natoľko neadekvátne, že vedie až k frustrácii používateľa [13]. O nič lepšie na tom nie je ani poskytnutie nadmernej funkcionality, pretože to má obyčajne negatívny vplyv na celkovú použiteľnosť aplikácie.

Definícia problému

Ako sa je možné dočítať už v úvode dokumentu, cieľom tejto bakalárskej práce je prevedenie stolovej hry do modernizovanej podoby s možnosťou jej **vylepšenia** bez straty kľúčových aspektov vyplývajúcich z jej hrania. Je známe, že na rozdiel od počítačových hier sa používatelia musia sústrediť priamo na ostatných, pevne ich spájajúc v spôsobe monitorovania objektov hracej plochy a akcií všetkých zúčastnených, čo vo väčšine prípadov vedie k stimulovaniu ich vzájomných sociálnych interakcií [6]. Podstatné pre návrh aplikácie je teda predovšetkým **zachovanie spoločenských aspektov** typických pre stolové hry, rovnako ako aj efektívne **virtualizovanie fyzických herných objektov** takým spôsobom, aby výsledné rozhranie malo čo najbližšie k originálu a zároveň bolo vďaka vhodnej vizualizácii so zrozumiteľnou interaktivitou dostatočne jednoduché na pochopenie i ovládanie.

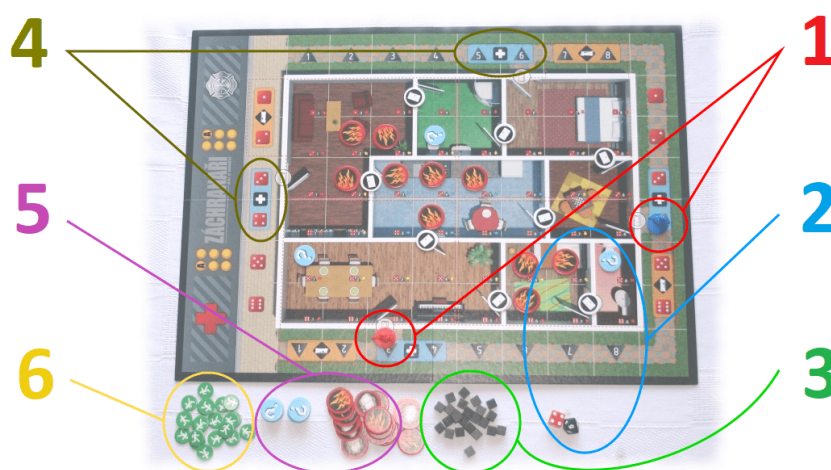
Spomínanými **vylepšeniami** oproti originálu sa samozrejme myslia v prvom rade **multimediálne prvky**. Spetrením audiovizuálnych podnetov pridaním grafiky s animáciami a zvuku akákoľvek hra naberá úplne nové rozmery v oblasti zážitnosti, čo vo výsledku vedie k výraznému zlepšeniu používateľskej skúsenosti. Hoci to však plánovaná aplikácia nepochybne umožňuje, nie je to primárnym cieľom jej prototypu. Ten sa totiž zameriava najmä na **zvýšenie dynamiky**, čo je okrem toho, že ide o relatívne netradičnú voľbu (vzhľadom na jej kooperatívnu povahu), najhlavnejším dôvodom, prečo bola na ilustráciu zdokumentovaných postupov vybraná práve zmieňovaná hra *Záchranári* od Mindok [15].

Z predchádzajúcej kapitoly je známe, že cieľom hry Záchranári je dostať čo najväčší počet ohrozených osôb z horiacej budovy. Jej potenciálne najväčším problémom je však skutočnosť, že pomerne komplikované šírenie ohňa a dopady akcií hráčov nie sú z pochopiteľných príčin v originálnej verzii automatizované. Povinnosť opakovane hádzať kockou na konci každého kola, sprevádzaná časovo náročným vyhodnocovaním na základe rozsiahlych pravidiel, kam sa majú presunúť plamene s dymom, aké sú následky rozrastania požiaru, kde všade sa prostredie vo vnútri budovy zmenilo a odkiaľ sa ozývajú novo objavené ohrozené osoby, po relatívne krátkej dobe prestáva byť zábavou. Nutnosť robiť to všetko manuálne je teda jedným z najväčších problémov, ktoré by aplikácia nepochybne mala riešiť.

Analýza herných prvkov a akcií

Aby bolo možné aplikáciu navrhnúť, je potrebné rozumieť všetkým komponentom originálnej hry Záchranári a súvisiacim akciám, ktoré je ktorýkoľvek hráč schopný v jej priebehu podniknúť. Práve z nich totiž vyplývajú nasledujúce problémy (pozri obrázok 3.1):

1. **vizualizácia hráčov** - všetci hráči musia byť reprezentovaní nejakým očividným objektom, ktorý je ovládateľný vždy iba vtedy, ak je dotýčný jedinec práve na ťahu
2. **šírenie dymu, ohňa a ohrozených osôb** - na konci každého kola je potrebné rozhodnúť o tom, kam sa rozšíri požiar a umiestnia novo objavení ohrození ľudia
3. **vizualizácia rozbitých stien a dverí** - na hracej ploche musí byť jednoznačne vidno všetky steny a dvere vrátane stavu ich poškodenia a možnosti prechodu
4. **záchranné stanice** - na mape musia byť rozmiestnené špeciálne vyznačené miesta, na ktoré sa hráči automaticky presunú v prípade, že budú zranení ohňom
5. **vizualizácia dymu, ohňa a ohrozených osôb** - objekty reprezentujúce ohrozené osoby alebo požiar potrebujú byť viditeľné (i pri vzájomnom prekrytí) a vymeniteľné
6. **zobrazovanie herných stavov** - herné stavy (napríklad hráč na ťahu, počet zostávajúcich akcií za kolo, poškodenie budovy, atď.) musia byť viditeľné a zrozumiteľné



Obr. 3.1: Ukážka všetkých základných komponentov hry Záchranári

Každý hráč na ťahu potrebuje mať umožnené vykonať ktorúkoľvek z povolených akcií, ktoré boli definované na konci predchádzajúcej kapitoly. V závislosti na nich a šírení požiaru aplikácia musí pravidelne náležito aktualizovať všetky herné stavy (vrátane informácií o aktuálnej pozícii záchranárov a ostatných elementov). V prípade, že dôjde na **ukončenie hry** v dôsledku usmrtenia príliš vysokého počtu uviaznutých osôb alebo zrútenia budovy, je nevyhnutné, aby o tom používatelia boli jasne informovaní. To isté samozrejme platí aj o **víťazstve**, pokiaľ sa zúčastneným podarí včas oslobodiť daný počet ohrozených ľudí.

Ďalšie požiadavky na aplikáciu

Existujú samozrejme aj nejaké ďalšie funkcie, ktoré stolová verzia hry Záchranári v rovnakej miere neposkytuje, ale navrhovaná aplikácia by ich ideálne obsahovať mala. Typickým príkladom je **možnosť nastavenia hry** pred jej spustením (počet hráčov, edícia, obťažnosť, atď.). Výsledný program by takisto mohol dovoliť **zobrazenie návodu**, rovnako ako aj **ukladanie a načítanie pozície (Save and Load)**. Pre pototyp to ale nie je nevyhnutné.

Vzhľadom na to, že aplikácia má byť spustiteľná s využitím interaktívneho stola definovaného v predchádzajúcej kapitole, je potrebné kalibrovať všetky jeho komponenty. Tento proces by mal byť ideálne taktiež automaticky riešený hneď po prvom otvorení výsledného programu, aby ho používateľ nebol nútený uskutočňovať manuálne (uberajúc si tak potenciálne na chuť k hraniu). Bolo by zároveň vhodné, aby to neprebiehalo pri každom jednom spustení aplikácie, ale len na začiatku (kalibrácia totiž zaberá zbytočne veľa času).

3.2 Herné stavy a funkcionality

Návrh programu na základe vyššie definovaných požiadaviek je vo veľkej miere koncipovaný pre desktopové prostredie. Hru by teda ideálne malo byť možné plne kontrolovať myšou a klávesnicou na počítači aj bez nutnosti vlastnenia akýchkoľvek dodatočných komponentov. Na splnenie cieľov tejto bakalárskej práce je však nevyhnutné, aby bolo možné spracovávať vstupy a výstupy i priamo s využitím interaktívneho stola. Až uplatnením konceptu **priesťorovej rozšírenej reality** spočívajúceho v zasadení virtualizovaného herného priestoru do skutočného prostredia bez nutnosti používateľov držať (alebo mať nasadené) akékoľvek zariadenie v kombinácii s ovládateľnosťou na báze **hmotných používateľských rozhraní** je totiž dodržaná požiadavka o zachovanie najkľúčovejších aspektov hrania stolových hier.

Na prevod originálu hry Záchranári do podoby spustiteľnej s využitím interaktívneho stola je nevyhnutné virtualizovať určité objekty a koncepty. To však nie je jednoduché vykonať bez podrobnejšej špecifikácie základných stavov a funkcionalít. Prvá časť návrhu sa teda venuje práve ich definovaniu a charakterizovaniu na základe vyššie uvedenej analýzy požiadaviek a pravidiel, počínajúc najhlavnejšími vlastnosťami jednotlivých herných fáz.

Fázy hry

Celý priebeh hry Záchranári je z kontinuálneho hľadiska pre jednoduchosť možné rozdeliť do troch najväčších logických celkov. Jedná sa v zachovanom poradí konkrétne o tieto:

1. **Začiatok hry:** Tento počiatočný stav spočíva primárne v nachystaní herných prvkov. Pozostáva z prípravy herného plánu, rozmiestnenia žetónov (dvere, oheň, ohrozené osoby) a polohy figúrok záchranárov na ľubovoľné štartovné pozície mimo vyznačenú budovu. Hru zahajuje podľa dohody ktokoľvek z hráčov prevedením prvej akcie.

2. **Striedanie akcií jednotlivých hráčov s následkami požiaru:** Každý hráč má k dispozícii štyri akčné body reprezentujúce, čo všetko môže počas svojho ťahu vykonať. Ako náhle ich všetky minie, dochádza k vývoju stavov budovy (rozšírenie požiaru s vyhodnotením následkov a doplnenie žetónov ohrozených osôb), ktorý zároveň ukončuje kolo a predáva kontrolu ďalšiemu v poradí (podľa dohody alebo v smere hodinových ručičiek). Takto to pokračuje až do doby, než je splnená jedna z podmienok ukončenia.
3. **Ukončenie hry:** Ak je zachránených minimálne sedem ohrozených osôb vynesением z nebezpečnej budovy, hráči víťazia. Pokiaľ sú naopak aspoň štyri usmrtené ohňom alebo zrútením celej konštrukcie v dôsledku šírenia požiaru, hráči prehrávajú. Oba tieto prípady vedú k okamžitému ukončeniu hry, ktoré spočíva akurát v informovaní o výsledku a uprataní herného plánu so všetkými využívanými objektmi.

Z poskytnutých informácií pre návrh vyplýva, že je nevyhnutné každú z týchto fáz viditeľne rozlíšiť, aby sa v nich používatelia boli schopní jednoducho orientovať. Zároveň je okrem samotného herného plánu so všetkými hernými objektmi potrebné vhodným spôsobom vizualizovať aj tie najkľúčovejšie indikátory priblíženia k ukončeniu hry (počet mŕtvych ohrozených osôb, poškodenie budovy a počet zachránených osôb), rovnako ako zmieňovaný počet zostávajúcich akcií, kto je aktuálne na ťahu, prípadne i o aké kolo sa jedná.

Herné akcie

Aplikácia musí každému hráčovi umožňovať vykonávať niekoľko rozličných akcií, ktoré vychádzajú z pravidiel a vyššie špecifikovaných požiadaviek. Na to, aby bolo možné určiť, akým spôsobom ich je vhodné zobrazovať a prevádzkať, je ich najskôr nutné kategorizovať:

- **Pohyb:** Hráč môže presunúť svojho záchranára na ľubovoľné susediace políčko výmenou za jeden akčný bod (v prípade, že prenáša ohrozenú osobu, za dva). Za susediace políčko sa však nepovažuje žiadne také, ktoré je od neho oddelené stenou, zavretými dverami alebo ohňom. V originálnom znení by malo byť možné postaviť sa i na políčko s ohňom výmenou za dva akčné body (nesmie tam ale záchranár zostať stáť na konci kola), no vzhľadom na to, že za rovnaký počet akcií je možné najskôr spomínaný oheň zmeniť na dym a až potom sa naň presunúť, bola táto funkcionálna pominutá.
- **Interakcia so stenami a dverami:** Záchranár je samozrejme schopný otvárať a zatvárať akékoľvek nepoškodené dvere výmenou za jeden akčný bod. Zároveň ale dokáže i strhávať steny výmenou za dva, pokiaľ sa niekam potrebuje dostať urýchlene (v originále ho to stálo dvojnásobok akcií, čo sa však neoplatilo, preto došlo k zmene).
- **Hasenie:** Každý hráč je schopný odstrániť jeden žetón dymu za akčný bod. V prípade, že sa chce zbaviť ohňa, ho najskôr musí premeniť na dym, čo ho stojí ďalšiu akciu.
- **Vynášanie ohrozenej osoby:** Záchrana uviaznutého jedinca nie je v návode dostatočne rozpísaná, preto bolo nutné do istej miery poupraviť a rozšíriť jej priebeh. Hráč musí v prvom rade odhaliť, či ohrozená osoba nie je planým poplachom (to spraví buď tým, že šlapne na miesto, kde sa cieľ nachádza, alebo to za akčný bod môže vykonať aj zo susedného políčka). Ak sa skutočne jedná o planý poplach, ohrozená osoba zmizne z hracej plochy. V opačnom prípade ju je možné buď ignorovať a vrátiť sa k nej neskôr (pokiaľ ju však medzitým zakryje dym, opäť bude zahalená, pretože nie je isté, či vôbec zostala stáť na mieste), alebo ju zodvihnúť a pohybom vyniesť z budovy (nie je možné prenášať viac než jednu naraz). Jedine až potom je zachránená.

Oproti stolovej verzii hry Záchranári je nutné pridať ešte niekoľko menej dôležitých funkcionalít, medzi ktoré patrí napríklad **návrat do úvodnej obrazovky** aplikácie. Za zmienku takisto stojí, že by po zahájení prenášania uviaznutého jedinca hráč mohol byť schopný kedykoľvek ho položiť a zasa zodvihnúť bez straty akčného bodu. Učiniť by to ale šlo iba na políčko bez akýchkoľvek prítomných žetónov. Nedáva totiž zmysel, aby dobrovoľne zostal stáť na zadymenom mieste, pričom polozenie viacerých ľudí na rovnakú pozíciu by zrejme nemalo byť ani vôbec povolené (nehovoriac o tom, že by to bolo veľmi neprehľadné).

Na základe *siedmich etáp činností používateľa* spomínaných v predchádzajúcej kapitole je zrejme, že rozhranie by malo v prvom rade vhodným spôsobom vizualizovať aktuálnu situáciu na hernom pláne (kde sa nachádza oheň, kde sú ohrozené osoby, či záchranár na ťahu prenáša niekoho iného, atď.), aby hráč nemal problémy s **formulovaním cieľa a zámeru**. Aplikácia takisto musí v rámci ulahčenia **špecifikovania akcie** zvýrazňovať, čo všetko je možné v danej chvíli vykonať. Samotné **prevedenie** nesmie byť zložité (pre potreby tejto hry by malo stačiť obyčajné stlačenie políčka alebo žetónu), pričom je opäť nevyhnutné, aby následky šlo jednoducho **vnímať, interpretovať a vyhodnotiť** (napríklad pri hasení ohňa musí byť vidno, že bol naozaj premenený na dym, čo by sa malo okamžite odzrkadliť i na zobrazenom počte zostávajúcich akcií a možnostiach ďalších interakcií).

Vývoj stavu budovy

Aplikácia oproti originálu automatizuje všetko dianie okrem samotného uskutočňovania akcií hráčov. Pre jednoduchosť ho je možné rozdeliť do nasledujúcich kategórií:

- **Rozšírenie požiaru:** Program by mal postupovať presne podľa pravidiel obsiahnutých v návode, ktorý je dostupný v prílohách a na oficiálnom zdroji [15]. Na základe pseudonáhodných čísel určí pozíciu nového dymu, ktorý je ihneď umiestnený. Pokiaľ na danom políčku už leží iný žetón dymu, zmení ho okamžite na oheň (to isté platí aj v prípade, že akurát priamo susedí s iným ohňom v rovnakej časti budovy vyhraničenej stenami). Ak je ale nový dym položený na políčko ohňa, dochádza k takzvanej explózii, ktorá sa prejaví rozmiestnením žetónov ohňa na všetky voľné a zadymené susediace políčka (ak natrafí na ďalší oheň, požiar sa posúva ešte ďalej). Pokiaľ explózia v nejakom smere nemôže nastať kvôli prítomnej stene alebo zatvoreným dverám, požiar zmiňované bariéry akurát poškodí (dvere sú vždy okamžite zničené, pričom steny je na ich úplnú demoláciu takýmto spôsobom nutné zasiahnuť až dvakrát).
- **Sekundárne efekty:** S rozšírením ohňa je úzko späté aj vyhodnocovanie, či novo pridané žetóny neusmrtili nejakú ohrozenú osobu. Ak tomu tak je, musí byť pochopiteľne odstránená z hracieho poľa, čo sa spoločne so škodami spáchanými na budove odrazí na základných stavoch hry (je sa preto nevyhnutné pravidelne uisťovať, či nebola splnená jedna z podmienok ukončenia). Všetky žetóny dymu, ktoré priamo susedia s ohňom, potrebujú byť navyše taktiež zmenené na oheň. Pokiaľ sa požiar rozšíril na políčko s niektorým z hráčov, záchranári sa presúvajú do najbližšej stanice s medikmi vyznačenej na mape (v originálnom znení môže dôjsť na voľbu, ale tá bola pominutá).
- **Rozmiestnenie nových ohrozených osôb:** Na konci každého kola sa majú v hre nachádzať tri žetóny ohrozených osôb. Ak tomu tak nie je, na základe pseudonáhodných čísel musia byť opäť zvolené pozície, kde budú umiestnení noví jedinci (môže sa ale jednať i o plané popluchy). V prípade, že prekryjú dym alebo oheň, majú prednosť (všetky ostatné žetóny na danom políčku sú jednoducho odstránené).

Ideálne by mali byť hráči schopní sledovať, čo všetko sa odohralo pri ukončení každého kola bez toho, aby to priveľmi zdržiavalo. Riešením môžu byť práve multimediálne prvky (predovšetkým atraktívne audiovizuálne efekty) a komplexnejšie animácie. Pre potreby tejto bakalárskej práce im však nie je nutné venovať čas, keďže ich vytvorenie by s najvyššou pravdepodobnosťou trvalo o dosť dlhšie ako samotné navrhnutie a pridanie.

3.3 Návrhové modely

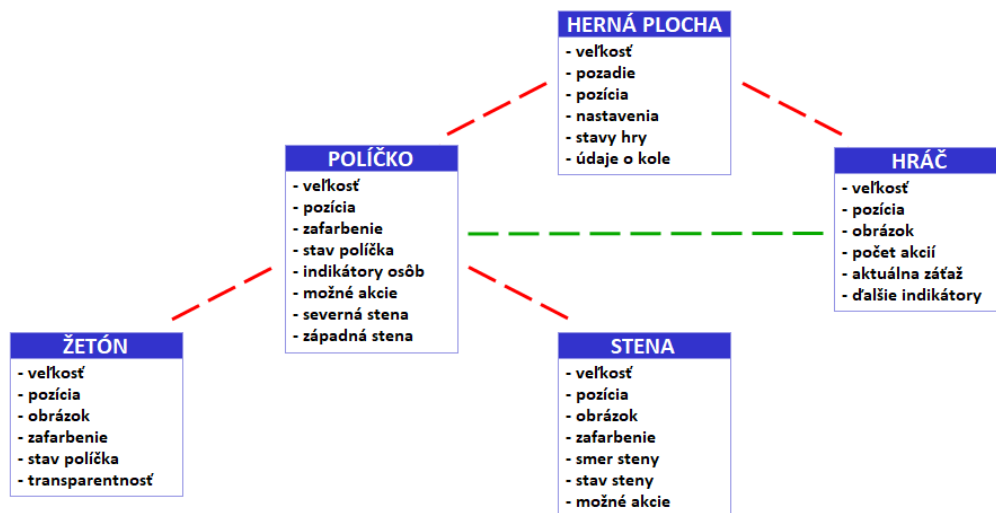
Na základe požiadaviek, pravidiel a bližšej špecifikácie funkcionality už je uskutočniteľný návrh konkrétnych herných objektov so stavmi, rovnako ako aj toho, akým spôsobom môžu byť reprezentované. Presne na to poslúži **dátový model**, sprevádzaný **procesným modelom**, ktorý je v tomto dokumente určený primárne na jasné uvedenie vyššie popísaných funkcionalít do kontextu s jednotlivými definovanými elementmi. Súčasťou finálnych krokov každého návrhu programu s grafickým používateľským rozhraním by malo byť taktiež i vytvorenie šablóny zobrazujúcej plánovaný dizajn, počínajúc jednoduchým náčrtom na papier vedúcim k jeho štrukturalizácii do podoby **drôteného modelu (wireframe)**.

Dátový model hry

Herné dáta, na základe ktorých dochádza k spracovaniu vstupov a výstupov, je možné štrukturalizovať do nasledujúcich objektov (pre orientáciu v hierarchii pozri obrázok 3.2):

- **Herná plocha:** Tento objekt je reprezentovaný štvoruholníkom, ktorý v sebe uchováva informácie o *velkosti*, *pozadí* (obrázok podľa edície hry) a *pozícii*. Zároveň obsahuje uložené *nastavenia* (obťažnosť, počet hráčov, edícia), pravidelne aktualizované *stavy hry* (počet mŕtvych, zachránených a poškodených častí budovy) a *údaje o aktuálnom kole* (hráč na ťahu, počet zostávajúcich akcií a záťaž, pokiaľ je prenášaná osoba).
- **Políčko:** Aj tento objekt je reprezentovaný štvoruholníkom, no uchováva v sebe okrem informácií o *velkosti* a *pozícii* taktiež *zafarbenie* (podľa toho, čo za akcie sa na ňom dajú vykonať, môže byť buď zelené, červené alebo úplne transparentné). Obsahuje i špecifický *stav políčka* určený na základe toho, či sa na ňom niečo nachádza (nič, dym, oheň alebo poplach), *indikátory osôb* (či na ňom nestojí hráč, ohrozená osoba alebo medik zo záchrannej stanice) a spomínané *možné akcie* (či ho je možné uhasiť, postaviť sa naň alebo identifikovať ohrozenú osobu). Na záver je nutné definovať aj priame prepojenie objektu s konkrétnymi **stenami** (stačí iba *severná* a *západná*).
- **Žetón:** Jedná sa o objekt zobrazujúci konkrétny *stav políčka* v podobe štvorca s *obrázkom* (dym, oheň alebo ohrozená osoba), ktorý má opäť nejakú *velkosť* a *pozíciu*. Pre prípad, že sa ich na sebe nachádzajú položených viac (napríklad osoba prekrytá dymom), musí mať špecifikovanú i premennú *mieru transparentnosti*. Rovnako ako **políčko** vyžaduje i *zafarbenie* (ak je to napríklad oheň, ktorý je možné uhasiť, musí byť na základe pozície **hráča** zrejmé, či tak môže v danú dobu učiniť).
- **Stena:** Tento objekt je vizualizovaný buď ako *obrázok* (pokiaľ ide o dvere) alebo čiara (pokiaľ ide o múr). Má každopádne definovanú *velkosť*, *smer* a *pozíciu* na základe nadradeného **políčka**. Podľa toho, aký má *stav* (žiadna, neprechodná, poškodená, zbúraná, zatvorená alebo otvorená), v sebe vyhodnocuje a uchováva vlastné *možné akcie* (otvorenie, zatvorenie alebo zbúranie). Ak je s ňou **hráč** niečo schopný vykonať, musí to byť opäť v závislosti na jeho pozícii zrejmé, preto obsahuje i *zafarbenie*.

- **Hráč:** Posledný hlavný objekt je reprezentovaný ako štvoruholník s definovanou *veľkosťou*, *pozíciou* a *obrázkom* záchranára (návrh je však koncipovaný takým spôsobom, že to je do budúca možné bez problémov zmeniť, aby bol úplne transparentný a viazaný na pozíciu fyzickej figúrky). Uchováva v sebe zároveň informácie o *počte zostávajúcich akcií* a zmieňovanej *záťaži* (podľa toho, či aktuálne prenáša nejakú osobu), ktoré sú v prípade, že je na ťahu, posielané **hernej ploche**. Môže samozrejme podľa potrieb obsahovať i *ďalšie indikátory* (napríklad to, aké má identifikačné číslo).



Obr. 3.2: Navrhnutý dátový model hry

Procesný model hry

Od spustenia hry musí aplikácia na základe jej nastavení prenesených z úvodnej obrazovky najskôr **nachystať všetky dátové objekty** (*hernú plochu* pokrytú *hráčmi* a *políčkami*, ktoré pozostávajú z náhodne rozmiestnených *žetónov* a podľa edície definovaných *stien*). Následne **umožňuje každému hráčovi rozhodnúť o štartovnej pozícii** a **spúšťa cyklus** dookola opakujúcich sa postupov. Jedná sa konkrétne o tieto (v zachovanom poradí):

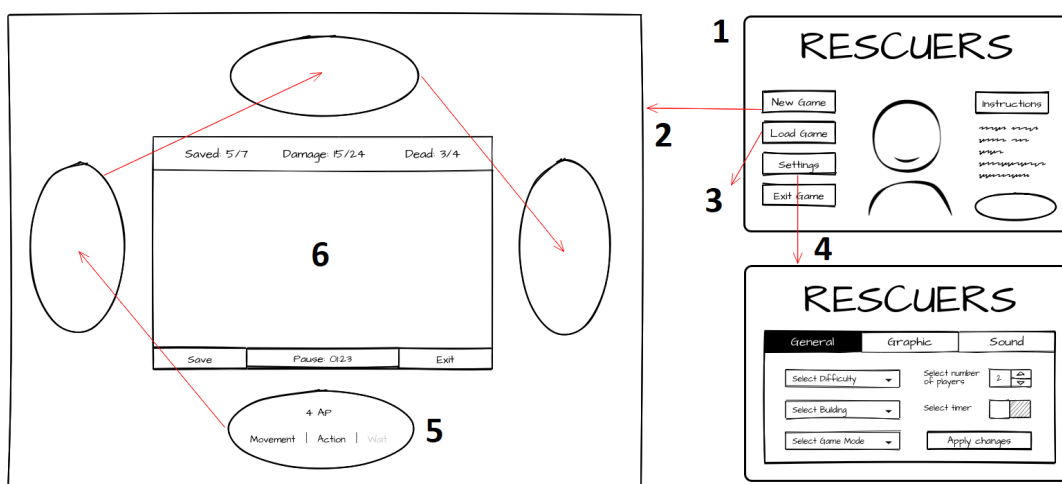
1. Aktualizácia zobrazovaných informácií podľa toho, ktorý hráč je práve na ťahu
2. Pridanie hráčovi adekvátneho počtu odohrateľných akcií (podľa pravidiel štyri)
3. Vyhodnotenie možných akcií hráča na ťahu, vyznačenie jeho postavy a zafarbenie všetkých objektov (políčok, žetónov a stien), s ktorými je schopný interagovať
4. Náležitá zmena všetkých potrebných stavov na základe zadaného vstupu hráča na ťahu (napríklad jeho pozície, ak sa presunul, alebo políčka a žetónu, ak chcel hasiť), pričom pokiaľ zámerne vynechal kolo, postupuje sa na začiatok procesu číslo 7
5. **Ukončenie hry**, ak hráč vykonal niečo, čo podľa pravidiel viedlo buď k víťazstvu (vynesenie poslednej ohrozenej osoby z budovy) alebo prehre (zbúranie budovy)
6. Odobratie adekvátneho počtu odohrateľných akcií hráčovi, pričom ak má ešte stále nejaké zostávajúce, postupuje sa naspäť na začiatok procesu číslo 3

7. Náhodné umiestnenie nového dymu a automatické rozšírenie ohňa podľa pravidiel
8. Náležitá zmena všetkých potrebných stavov na základe rozšírenia ohňa (napríklad stien, susediaceho dymu, života ohrozených osôb alebo pozície hráča, pokiaľ bol zranený), pričom ak to podľa pravidiel viedlo k prehre, dochádza k **ukončeniu hry**
9. Umiestnenie nových ohrozených osôb podľa pravidiel, prepnutie hráča na ťahu (na ďalšieho v poradí), zmena aktuálneho kola a návrat na začiatok procesu číslo 1

Drôtový model aplikácie

Vzhľadom na to, že približný vzhľad samotnej mapy hry pokrýva originál, je v kontexte tejto bakalárskej práce pri vytváraní drôtového modelu potrebné zamerať sa iba na ostatné pridané prvky. Pozostáva teda len z nasledujúcich elementov (pozri obrázok 3.3):

1. **úvodná obrazovka** - hlavné okno programu obsahujúce tlačidlá so základnými funkciami (nová hra, načítanie pozície, nastavenia, zobrazenie návodu a ukončenie)
2. **herná obrazovka** - okno určené na samotné hranie, ktoré po spustení novej hry (prípadne načítaní uloženej pozície) prekryje úvodnú obrazovku
3. **selektor uložených pozícií** - jednoduchý výber konkrétnej uloženej pozície (do prototypu aplikácie ho nie je nevyhnutné zakomponovať)
4. **obrazovka nastavení** - okno s nastaveniami (obťažnosť, hráči, edícia, atď.), ku ktorému je možné prostredníctvom daného tlačidla pristúpiť z úvodnej obrazovky
5. **ovládací panel** - vymedzený priestor umožňujúci každému hráčovi ovládať svoje postavy (v priebehu vývoja bol napokon pre neprehľadnosť kompletne odstránený)
6. **hracia plocha** - mapa hry Záchranári s hernými prvkami, tlačidlami (uloženie pozície, pozastavenie a ukončenie hry) a zobrazenými pravidelne aktualizovanými stavmi (ubehnutý čas, poškodenie budovy, počet zachránených jedincov a úmrtí, atď.)



Obr. 3.3: Drôtový model (wireframe) aplikácie

Kapitola 4

Popis implementácie

Táto kapitola oboznamuje čitateľa s aplikovanými metódami implementácie programu navrhnutého v predchádzajúcej časti dokumentu. Obsahuje zoznam všetkých nástrojov i programovacích jazykov použitých pri vývoji aplikácie, stručný náhľad na každý z najdôležitejších prvkov používateľského rozhrania vrátane vzťahov medzi nimi, vysvetlenie funkcionality predovšetkým z hľadiska hernej logiky a popis spracovania vstupov snímaných z interaktívneho stola, sprevádzaných projekciou výstupu v podobe obrazu na jeho povrch.

4.1 Implementačné nástroje

Program bol implementovaný vo vývojom prostredí *QT Creator* s multiplatformným aplikačným rámcom **Qt 5**. Jeden z hlavných dôvodov tohto výberu je integrácia deklaratívneho jazyka **QML**, určeného na popis používateľských rozhraní s využitím stromovej štruktúry objektov a mnohých konštrukcií **JavaScriptu** [16]. Vďaka tomu zvolené prostredie poskytuje obrovskú kontrolu nad výsledným výzorom aplikácie, čo je v kombinácii s možnosťou jednoduchého vytvorenia *balíka QT ROS* (kľúčového k spozojdnaniu riadenia komunikácie medzi komponentmi interaktívneho stola) pre potreby tejto bakalárskej práce ideálne.

Využitie programovacích jazykov

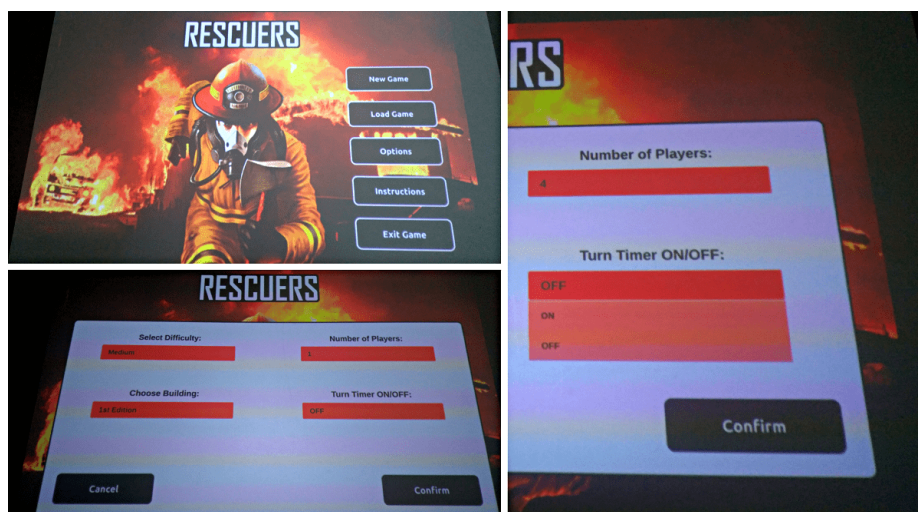
Vrstva aplikácie, ktorá zaobstaráva automatickú kalibráciu komponentov interaktívneho stola vrátane ich zmieňovanej vzájomnej komunikácie a samotného spustenia hlavného okna, je implementovaná v skriptovacom jazyku **Python**. Na umožnenie jeho využívania v Qt 5 slúžia rozširujúce moduly *PyQt5*, nahrádzajúce funkcionality mnohých **C++** knižníc [11].

Grafické používateľské rozhranie aplikácie je popísané v súboroch so zdrojovým kódom v jazyku **QML**. To zahŕňa úplne všetky jeho prvky, od úvodnej obrazovky s tlačidlami cez nastavenia až po samotnú hraciu plochu, ktoré sú tvorené objektmi typov definovaných v moduloch *Qt Quick*. Za zmienku takisto stojí, že práve v týchto súboroch sú importované funkcie reprezentujúce procesy z navrhovaného modelu, nachádzajúce sa v samostatnom skripte napísanom v jazyku **JavaScript**. Úplne všetky spomínané implementačné nástroje teda zohrávajú kľúčovú rolu v realizácii vybranej hry Záchranári pre interaktívny stôl.

4.2 Úvodná obrazovka s nastaveniami

Dizajn hlavného okna je veľmi podobný originálnemu drôtovému modelu z návrhu (pozri obrázok 4.1). Všetky žiadané prvky tvoriace úvodnú obrazovku a spustiteľnú plochu s nastaveniami je na základe funkcionality možné kategorizovať nasledujúcim spôsobom:

- **Okno úvodnej obrazovky:** Tento prvok so zdrojovým kódom v *TitleScreen.qml* je z hierarchického hľadiska *QML* možné považovať za koreňový. Jedná sa o objekt typu *Window* s niekoľkými základnými funkciami (spustenie, vypnutie, atď.), obrázkom na pozadí a **tlačidlami**. Obsahuje taktiež aj špeciálny mód prekrytia čiernou plochou so zobrazovanými bielymi bodmi, ktoré je nutné stlačiť pri kalibrácii dotykovej plochy.
- **Tlačidlo:** Daný element definovaný v *MenuButton.qml* je objektom typu *Rectangle*, ktorý zohráva rolu tlačidla využívaného na úvodnej obrazovke a **ploche s nastaveniami**. Jednotlivé operácie spúšťa volaním funkcie *executeMenuButton()* popísanej v **okne úvodnej obrazovky**, ktorá na základe predaného argumentu *operation* prostredníctvom jednoduchých podmienok vyhodnotí a prevedie, čo sa má vykonať.
- **Chybová správa:** Tento prvok popísaný v *ErrorMessage.qml* tvorí objekt typu *Rectangle*, ktorý v grafickom používateľskom rozhraní aplikácie (i na hernej obrazovke) zohráva rolu dialógového okna s popisom chyby alebo upozornením vyplývajúcim zo vstupu používateľa. Prekrýva úplne celú obrazovku programu, aby nebolo možné pred jeho vypnutím kliknúť na nič iné, no iba menší úsek so správou nie je transparentný.
- **Plocha s nastaveniami:** Základom daného prvku so zdrojovým kódom v súbore *SettingsScreen.qml* je objekt typu *Rectangle*. Pri spustení prekrýva pomerne veľkú časť úvodnej obrazovky. Pozostáva z **tlačidiel** na jej vypnutie (buď s uložením zmien alebo bez) a výberových prvkov reprezentujúcich nastavenia (edícia, obťažnosť, atď.).
- **Výberový prvok nastavení:** Tento element definovaný v súbore *SettingsComboBox.qml* tvorí objekt typu *Rectangle*. Zohráva rolu **rozbaľovacieho zoznamu** (*drop-down list*), k čomu využíva štvoruholník s *ListView* a tranzíciu s *NumberAnimation*.



Obr. 4.1: Ukážka úvodnej obrazovky a plochy s nastaveniami premietanej na ARTable3

4.3 Herná obrazovka

S pominutím pôvodne navrhovaných ovládacích panelov, ktoré zaberali priveľa miesta a činili hru menej prehľadnou, dizajn hernej obrazovky taktiež vychádza z drôtového modelu (pozri obrázok 4.2). Jeho prvky sú podľa funkcionality kategorizované týmto spôsobom:

- **Plocha hernej obrazovky:** Z hierarchického hľadiska *QML* je v stromovej štruktúre najvyššie postaveným základným prvkom (s výnimkou koreňového) tento objekt typu *Rectangle*, definovaný v *GameScreen.qml*. Po stlačení špecifického tlačidla úvodnej obrazovky (New Game) pokrýva hlavné okno. V návrhu dátového modelu je referovaný ako *herná plocha*. Skladá sa z **herných tlačidiel**, pravidelne aktualizovaného textu, časomieru typu *Timer*, **indikátorov života** a nižšie špecifikovanej mapy.
- **Herné tlačidlo:** Tento element popísaný v *GameButton.qml* tvorí objekt typu *Rectangle*, ktorý zohráva rolu tlačidla využívaného výhradne iba na hernej obrazovke. Jednotlivé operácie spúšťa volaním funkcie *executeGameButton()*, ktorá sa od vyššie zmieňovanej *executeMenuButton()* líši iba rozdielnou definíciou požadovaných úkonov.
- **Herný štvorec:** Daný prvok so zdrojovým kódom v *GameSquare.qml* pozostáva primárne z objektu typu *Rectangle*, ktorý podľa návrhu zohráva rolu *políčka* a *žetónu*.
- **Herná stena:** Tento element definovaný v *GameWall.qml* je tvorený z veľkej časti transparentným objektom typu *Rectangle*, ktorý pokrýva *políčko*. Jedná sa o priameho potomka **herného štvorca**, nazývaného v návrhu dátového modelu ako *stena*.
- **Indikátor života:** Základom tohto prvku popísaného v *HealthBar.qml* sú dva prekrývajúce sa objekty typu *Rectangle* (jeden je nadradený) rôznych farieb a veľkostí. Daná štruktúra im umožňuje poskytovať funkcionality ukazovateľa zdravia, ktorý je využitý na vizualizáciu poškodenia budovy, počtu úmrtí a počtu zachránených osôb.
- **Záchranár:** Tento prvok so zdrojovým kódom v súbore *Player.qml* je objektom typu *Rectangle*, reprezentujúcim podľa návrhu dátového modelu *hráča*. Obsahuje i behaviorálne animácie pohybu *NumberAnimation*. Podrobnejšie je tiež rozobratý až nižšie.



Obr. 4.2: Ukážka hernej obrazovky premietanej na ARTable3

Herná mapa

Na inicializáciu herného plánu je v prvom rade potrebné deklarovať a definovať niekoľko premenných uchovávajúcich informácie vychádzajúce z pravidiel hry (napríklad maximálny počet ohrozených osôb za jedno kolo, počet políček, atď.). To obstaráva funkcia *startGame()* importovaná zo súboru **gameboard.js**, ktorá zároveň naplní jednorozmerné pole *board[]* dynamicky vytvorenými QML objektmi **herných štvorcov** volaním *createSquare()*. Ich steny sú v zápätí definíciou vybraných stavov prostredníctvom *createWalls()* určené na základe zvolenej edície Záchranárov (každá ich má totiž po budove rozmiestnené inak). Inicializovaná mapa zároveň v jednorozmernom poli *medics[]* obsahuje aj informácie o pozíciách záchranných staníc dané funkciou *createMedics()* podľa vopred stanovených pravidiel.

Okrem samotných políček bolo nevyhnutné nejakým spôsobom modelovať zmieňované steny. Každé jedno z nich má na to určených dvoch potomkov, a to severnú a západnú **hernú stenu** (ich stavy sú reprezentované atribútmi *topWall* a *leftWall*), ktoré sa môžu chovať a vyzeráť buď ako žiadne steny, neprechodné steny, otvorené dvere, zatvorené dvere, poškodené steny alebo rozbité steny či dvere. Výzor a vlastnosti celého políčka určujú zasa stavy **herného štvorca**, na základe ktorých je vyhodnocovaný prináležiaci žetón (okrem dymu a ohrozenej osoby sa žiadne dva nikdy neprekrývajú, takže jediným špeciálnym prípadom je zadymený poplach daný atribútom *smokedAlert*). Môže ísť konkrétne o prázdny žetón, dym, oheň, poplach (otáznik potenciálne skrývajúci ohrozenú osobu) a odhalenú alebo zadymenú ohrozenú osobu (všetky tieto stavy sú reprezentované atribútom *state*).

S každým jedným spomínaným elementom (políčko, žetón, stena) je používateľ pri splnení podmienok uvádzaných v návrhu schopný externe interagovať. Na to, aby však mohol posúdiť, kedy a za akých okolností, bol pre všetky modelovaný atribút zafarbenia (napríklad transparentné, zelené, červené, atď.), ktoré po každom prevedenom čine záchranárov v závislosti na pozícii hráča na ťahu, žetónov a stavov stien mení s využitím niekoľkých zložených podmienok funkcia *enableAvailableSquares()*. Práve tá určuje možné akcie realizovateľné fyzickým kontaktom s vyznačenými prvkami, ktoré vedú k reakcii vychádzajúcej z pravidiel (napríklad ak je to dostupné políčko so žetónom ohňa, dochádza na volanie *extinguishFire()*, ktoré ho zmení na dym). Vďaka tomuto konceptu nie je nevyhnutné, aby hráči pracovali so zvažovaným kontextovým menu alebo ovládacím panelom, keďže na každý element mapy hry s konkrétnou kombináciou stavov sa vždy viaže iba jediná povolená akcia.

Hráči

Objekty **záchránárov** sú dynamicky vytvorené a umiestnené do jednorozmerného pola *players[]* v rámci funkcie *startGame()* importovanej zo súboru **gameboard.js**, a to na základe nastavenia počtu hráčov na úvodnej obrazovke reprezentovaného atribútom *players*. Nemôžu však na mapu byť položené náhodne, pretože je nevyhnutné umožniť používateľom vybrať si štartovné pozície mimo interiéru budovy. To obstaráva funkcia *firstMove()*, ktorá zvýrazní všetky povolené políčka (konkrétne internou zmenou ich zafarbenia na zelenú). Stlačením ktoréhokoľvek zo zvýraznených miest dochádza na korektné umiestnenie hráča.

Samotný návrh aplikácie je koncipovaný takým spôsobom, aby bolo možné pomerne jednoducho objekt **záchránára** urobiť transparentným a viazať ho na pozíciu skutočnej figúrky. V súčasnosti však je reprezentovaný obrázkom daným atribútom *source*, ktorého vzájomné prekrývanie so žetónmi je riešené priebežným prispôbovaním jeho priehľadnosti (atribút *opacity*). Stavy a akcie, ktoré môže hráč vykonať, sú špecifikované v predchádzajúcich kapitolách, pričom všetky spočívajú v interakcii s konkrétnym prvkom hernej mapy.

V prípade, že je v dôsledku rozšírenia ohňa hráč zranený, automaticky musí byť presunutý do najbližšej záchranej stanice. K tomu je potrebné vypočítať vzdialenosť od každého medika a určiť toho správneho, na čo slúži funkcia *findClosestMedicTo()* s návratovou hodnotou reprezentujúcou pozíciu zvoleného políčka. Samotné presunutie zabezpečuje interne *hurtPlayer()*. Za zmienku ešte stojí, že by sa podľa pravidiel hry každý hráč mal byť schopný okrem akcií definovaných v návrhu postaviť aj na políčko so žetónom ohňa výmenou za dve prevedené akcie. Táto možnosť ale nebola vo finálnej verzii programu implementovaná, pretože pre základnú verziu hry postráda zmysel, pokiaľ nie je úmyslom daného jedinca zámerne sa zraniť požiarom (čo by z pochopiteľných dôvodov nemalo byť povolené).

Herná logika

Väčšina funkcií definujúcich hernú logiku sa nachádza v súbore **gameboard.js** napísanom v skriptovacom jazyku *JavaScript*. Hra začína vyššie popisovanou inicializáciou mapy a hráčov s využitím *startGame()*, ktorá je nasledovaná náhodným rozmiestnením vybraného počtu žetónov (potenciálne ohrozené osoby a oheň). Po sprevádzanom určení štartovnej pozície záchranárov prostredníctvom *firstMove()* dochádza na cyklus popísaný v procesnom modeli z predchádzajúcej kapitoly, ktorý je možné rozdeliť do týchto nasledujúcich funkcionálit:

- **Vyznačenie herných prvkov s možnosťou interakcie:** Táto funkcionálna riešená *enableAvailableSquares()* spočíva v zrušení vyznačenia všetkých herných prvkov prostredníctvom *disableAllSquares()* a zmene zafarbenia nových podľa hráča na ťahu v závislosti na mape cez zložené podmienky definované vo *findAvailableSquares()*.
- **Pohyb záchranára:** Túto funkcionálnu zabezpečuje *moveCurrentPlayer()*. Spočíva v prvom rade v internej zmene pozície aktuálneho hráča na ťahu podľa zadaného vstupu a následnom prevádzaní náležitých interných akcií (odobratie akčných bodov s možnosťou ukončenia hry pri víťazstve prostredníctvom *finishAction()*, nastavenie vybraných herných stavov a prevedenie prípadných podružných operácií, ako je napríklad odhalenie ohrozenej osoby stojacej na rovnakom mieste cez *revealAlert()*).
- **Pohnutie dverami:** Jedná sa o prosté otvorenie alebo zatvorenie dverí definované funkciou *moveDoors()* na základe pozície aktuálneho hráča na ťahu a zadaného vstupu internou zmenou stavu vybranej steny (atribút *state*), ktoré končí volaním funkcie *finishAction()* (tá slúži primárne na správne odobratie akčných bodov záchranára).
- **Ostatné akcie záchranára:** Funkcia *playerAction()* vyhodnocuje možné akcie podľa povoleného zadaného vstupu, čo môže viesť napríklad k haseniu ohňa s využitím *extinguishFire()*, búranie steny definovaného v *damageWall()* alebo odhaleniu susediacej ohrozenej osoby cez *revealAlert()*. Vždy každopádne končí zmenou potrebných stavov herného plánu a odobratím odohrateľných akcií pomocou volania *finishAction()*.
- **Ukončenie kola:** Túto akciu vykonáva funkcia *finishTurn()* pridaním odohrateľných akcií aktuálnemu hráčovi do ďalšieho kola cez *addActionPoints()* a zmenou herných stavov (vrátane hráča na ťahu a súčasného kola) prostredníctvom *switchPlayer()*. Tá ďalej volá ešte aj funkciu *addSmoke()* na šírenie ohňa, ktoré je špecifikované nižšie, sprevádzanú **checkAfterEffects()** na kontrolu následkov rozrastajúceho sa požiaru.

- **Šírenie požiaru:** Spomínané šírenie ohňa spočíva v prvom rade v pridaní dymu na pseudonáhodnú pozíciu (*Math.random()*) prostredníctvom *addSmoke()*. Nasleduje zistenie okolitého dymu cez *checkSmokeAround()* a prekrytie ohrozenej osoby (pokiaľ sa na tom mieste nejaká nachádza). V prípade, že výsledky automatického vyhodnotenia stavov hernej mapy podľa pravidiel ukazujú na nejakú reťazovú reakciu, dochádza k volaniu *createFire()* na pridanie ohňa či *createExplosion()* na napodobenie explózie.
- **Zvyšné následky ukončenia kola:** Všetky ostatné následky zabezpečuje funkcia *checkAfterEffects()* ich realizáciou podľa pravidiel (zmena dymu susediaceho s ohňom na plamene cez *checkFireAround()*, usmrtenie zasiahnutých ohrozených osôb alebo zranenie záchranára prostredníctvom *hurtPlayer()*). To zahŕňa i pridanie nových ohrozených osôb s využitím funkcie *addAlert()* a ukončenie hry prehrou v prípade, že došlo na zrútenie budovy v dôsledku poškodenia stien alebo prekročenia limitu mŕtvych.

Program samozrejme obsahuje aj niekoľko ďalších funkcionalít (napríklad **položenie prenášanej ohrozenej osoby** riešené prostredníctvom *unloadAlert()*, **overenie susedstva daného miesta so žiadaným políčkom** pomocou *isNeighbor()*, **získanie hodnoty reprezentujúcej vzdialenosť hráča na ľahu od špecifickej záchranej stanice** prevádzané funkciou *getDistance()*, atď.). Za zmienku však stojí už len **samotné ukončenie hry** obstarávané volaním funkcie *gameOver()*, ktoré pozostáva z vypísania oznamu o výhre alebo prehre (aktuálne je k tomu využívaný spomínaný objekt **chybovej správy**), znemožnenia ďalšieho zadávania vstupov v oblasti hernej mapy a zastavenia časomier.

4.4 Integrácia do robotického operačného systému

Realizácia aplikácie s využitím interaktívneho stola je závislá na vzájomnej komunikácii všetkých jeho komponentov v reálnom čase. Veľká časť výsledného programu sa teda musí zaoberať práve jej sústavným riadením, na ktoré je pre potreby tejto bakalárskej práce využitý **robotický operačný systém (Robot Operating System, ROS)**. Pre prehľadnosť však zmieňovaná komunikácia prebieha výhradne iba na pozadí a do samotného grafického používateľského rozhrania hry priamo nezasahuje (všetky jej potrebné výpisy sú posielané len v textovom formáte na štandardný výstup dostupný cez príkazový riadok).

Kalibrácia jednotlivých komponentov interaktívneho stola (konkrétne dotykovej plochy a projektoru) je na základe definovaných požiadaviek automatizovaná pri prvom spustení programu. Ako náhle je úspešne prevedená, aplikácia sníma vstupy z povrchu stola (v aktuálnej verzii len prostredníctvom fyzického kontaktu používateľa s dotykovou plochou) a reaguje v závislosti na jasne daných pravidlách premietaním náležitého výstupu na interaktívny stôl prostredníctvom nepretržitého posielania aktualizovaného obrazu projektoru.

Spustenie aplikácie na interaktívnom stole

Na umožnenie čo najjednoduchšej integrácie aplikácie do robotického operačného systému pre naviazanie vzájomnej komunikácie všetkých komponentov interaktívneho stola AR-Table3 je potrebné v prvom rade vytvoriť balík v rámci **pracovného prostredia (workspace)** prekladového systému *catkin* [10]. Riešenie tohto problému bolo do značnej miery inšpirované staršou verziou existujúceho balíka s názvom *art_projected_gui*¹, na základe ktorého boli okrem iného aj manuálne vytvorené tri súbory uvedené na nasledujúcej strane.

¹Balík *art_projected_gui*, https://github.com/robofit/artable/tree/master/art_projected_gui

- **package.xml**: Tento súbor poskytuje všetky dôležité metainformácie o balíku (vlastnosti ako napríklad názov, meno autora, udržiavateľ, závislosti na iných balíkoch, atď.).
- **CMakeLists.txt**: Tento súbor je vstupom prekladového systému *CMake* popisujúcim, akým spôsobom a kam má byť nainštalovaná výsledná aplikácia. Využíva okrem iného *catkin_python_setup()*, *roslint_python()* a *roslint_add_test()*.
- **setup.py**: Súbor v jazyku *Python* nastavuje argumenty, na základe ktorých prekladový systém ROS s využitím *catkin_python_setup()* inštaluje výslednú aplikáciu.

Na to, aby bol program plne funkčný na interaktívnom stole ARTable3, je pred započatím jeho behu (*roslaunch xbacov04 rescuers_gui.py*) nevyhnutné v prvom rade uistiť sa, že je spustený *roscore*. V opačnom prípade ho je nutné aj s projektorom zapnúť, rozmiestniť štyri hlavné **značky (markery)** dostupné v laboratóriu RoboLab VUT FIT podľa ich textového popisu na povrch dotykovej plochy a zadaním "*roslaunch art_bringup bringup.launch*" do príkazového riadka *Ubuntu* (nainštalovaného na počítači prepojenom s ostatnými komponentmi interaktívneho stola) spustiť všetky dôležité súčasti systému (ovládače, uzly, atď.). Po následnom odložení zmieňovaných značiek je už aplikácia za predpokladu, že bola správne nainštalovaná, definitívne pripravená na uvedenie do chodu.



Obr. 4.3: Ukážka rozmiestnenia značiek na interaktívnom stole

Kalibrácia projektoru

Automatická kalibrácia projektoru po rozbehnutí aplikácie spočíva v prvom rade vo vytvorení servera scény, na ktorý sa pripája **uzol (node)** spustený balíkom *art_bringup*. Následne program inicializuje inštanciu triedy **ProjectorCalibrator** definovanej v zdrojovom súbore *projector_calibrator.py*, prostredníctvom ktorej cez vybrané **témy (topics)** posielaním a prijímaním **správ (messages)** komunikuje s kalibrovaným komponentom interaktívneho stola. Zavolanie **servisu (service)** projektoru s názvom *calibrate* typu **Trigger** (importovaného z balíka *std_srvs* [10]) už zaobstará všetko ostatné, akurát predtým musí uzol aplikácie na jednu zo spomínaných tém poslať správu typu **Bool** (importovaného z balíka *std_msgs* [10]) s obsahom **False** a po skončení kalibrácie naopak s obsahom **True**.

Kalibrácia dotykovej plochy

Po završení kalibrácie projektoru, kedy je už možné spracovaný výstupný obraz v reálnom čase premietat na interaktívny stôl, dochádza ihneď k príprave dotykovej plochy na použitie. Na to aplikácia inicializuje inštanciu triedy **TouchCalibrator** definovanej v zdrojovom súbore *touch_calibrator.py*, prostredníctvom ktorej opäť dochádza ku komunikácii cez vybrané **témy (topicy)**. Nestačí však už iba zavolať **servis (service)** dotykovej plochy s názvom *calibrate* typu **EmptySrv** (importovaného z balíka *std_srvs* [10]). Ten totiž predtým potrebuje ešte iný servis typu **TouchCalibrationPoints** (importovaného z balíka *art_msgs* [10]) so zoznamom súradníc bodov, ktoré budú premietnuté na interaktívny stôl.

Na zobrazenie daných bodov program využíva špeciálny mód grafického používateľského rozhrania úvodnej obrazovky, kde dochádza na jej kompletne prekrytie čiernou plochou. Tá má stavy reprezentujúce súradnice, s ktorými je jednoduché vykresliť biele kruhy na povrch interaktívneho stola (je ich však nutné prepočítať z metrov na pixely a naopak). Prijímaním **správ (messages)** cez tému s názvom *touch_detected* aplikácia detekuje stlačenie zmieňovaných bodov a vysiela signál vedúci k udalosti, ktorá zobrazí zasa ďalšie body. Takto to pokračuje až do doby, než má servis *calibrate* dostatočne veľa získaných údajov na to, aby mohol dotykovú plochu s relatívne vysokou presnosťou skalibrovať podľa projektoru.

Spracovanie vstupov a výstupov

Ako bolo povedané už vyššie, vrstva aplikácie zaobstarávajúca automatickú kalibráciu komponentov interaktívneho stola, ich vzájomnú komunikáciu a samotné spustenia hlavného okna je implementovaná v skriptovacom jazyku **Python**. Jedná sa konkrétne o zdrojový súbor s názvom *rescuers_gui.py*. Zmieňovanou komunikáciou sa v tomto prípade myslí aj spracovanie vstupov z povrchu dotykovej plochy a výstupov posielených projektoru.

Spomínané vstupy sú reprezentované ako správy typu **Touch** (importovaného z balíka *art_msgs*), ktoré program obdržiava vďaka obsluhu udalostí odoberaním **témy (topicu)** s názvom *touch*. Vďaka tomu má aplikácia takmer okamžitý prístup ku všetkým súradniciam dotyku, ktoré je akurát nutné prepočítať na pixely a vytvoriť s ich pomocou vlastné udalosti kliknutia myšou. Tie stačí pravidelne posilať koreňovému objektu používateľského rozhrania a hra sa stáva ovládateľná nielen na počítači, ale i priamo na interaktívnom stole.

Ako výstup program v stanovených časových úsekoch posiela na vyššie zmieňovaný server scény pole bajtov. Tie tvoria obrázok formátu *JPG*, ktorý je výsledkom konverzie a zrkadlového horizontálneho otočenia obrazovky grafického používateľského rozhrania. O technických detailoch sa je možné dočítať v oficiálnej referenčnej príručke k *PyQT5* [11] a zdrojovom súbore *rescuers_gui.py*, ktorý je súčasťou pamäťového média v prílohách.

Kapitola 5

Testovanie aplikácie

V tejto kapitole sa je možné dočítať o prevedenom priebežnom a finálnom testovaní výsledného programu. Zaoberá sa v stručnosti kontrolou funkcionality, no predovšetkým zisťovaním a vyhodnocovaním používateľskej spokojnosti na základe nižšie popísaného plánu. Záver tejto časti čitateľom poskytuje všetky najdôležitejšie výsledky testovania vrátane vyplývajúcich návrhov na vylepšenia a prípadné ďalšie rozšírenia zhotovenej aplikácie.

5.1 Funkcionálne testovanie

Výskum kvality aplikácie bol vzhľadom na časové nároky a obmedzené prostriedky v priebehu vývoja uskutočňovaný len manuálne po pridaní každej novej funkcionality, pričom finálne vyhodnotenie jej **funkčnosti** bolo zámerne prevedené v dostatočnom predstihu oproti **použitelnosti**. Z kapacitných dôvodov ale nedošlo na dodržanie úplne všetkých šiestich krokov funkcionálneho testovania definovaných v druhej kapitole. Hra bola v poslednej fáze vývoja akurát niekoľkokrát prejdená za účelom vyskúšania primárnych možností používateľov a odhalenia čo najväčšieho počtu kritických chýb. V konečnom dôsledku došlo na opravu najmä sémantických a logických nedostatkov (hru šlo napríklad aj po jej ukončení stále pozastavovať a obnovovať, čo viedlo k nechcenému volaniu určitých funkcií). Vzhľadom na to, že sa však program po eliminovaní niekoľkých drobností zdanlivo choval presne podľa očakávaní, bol o dosť väčší dôraz kladený až na samotné **používateľské testovanie**.

5.2 Plán používateľského testovania

Cieľom tohto testovania je vyhodnotenie **použitelnosti** výslednej aplikácie a očakávanej používateľskej spokojnosti na základe spätnej väzby účastníkov. Výstupom by malo byť odhalenie najväčších problémov týkajúcich sa predovšetkým dizajnu so zameraním na **účelovosť** (v tomto prípade hrateľnosť), **efektívnosť** (jednoduchosť a ovládateľnosť), **užitočnosť** (do akej miery aplikácia poskytuje funkcionality potrebnú pre hranie), **naučiteľnosť** (jak jednoduché je pochopenie hry), **zapamätateľnosť** (náročnosť zapamätania si všetkých herných a ovládacích prvkov) a **bezpečnosť** (najmä ochrana pred nechcenými akciami).

Plán používateľského testovania je rozdelený do týchto logických celkov:

- **Analýza cieľov testovania** - sumarizácia cieľov aplikácie, na ktoré sa treba sústrediť
- **Cieľová skupina** - stručný popis používateľov, na ktorých je hra zameraná, vrátane krátkych obecných profilov jedincov vybraných na testovanie

- **Metodológia** - popis úvodného oboznámenia účastníkov testovania s aplikáciou, vysvetlenie rolí a procedúra (počet iterácií, dĺžka, prostredie, postup)
- **Metódy hodnotenia** - zoznam metrík, dotazník (príloha) a očakávané chyby

Analýza cieľov testovania

Cieľom hry Záchranári je dostať čo najväčší počet osôb z horiacej budovy skôr, než sa požiar rozšíri dostatočne na to, aby všetkých jej obyvateľov usmrtil. Jedná sa o kooperatívnu zábavu, čo znamená, že hráč by mal spolupracovať s ostatnými, aby dospel k požadovanému záveru. V každom kole môže uskutočniť štyri akcie, a to buď **pohyb**, **hasenie**, **otváranie** či **zatváranie** dverí, **búranie** stien, **vynášanie** zranených jedincov alebo **vynechanie** ťahu.

Hoci je program ovládateľný aj bez interaktívneho stola, jednou z jeho kľúčových požiadaviek je **zachovanie sociálnych aspektov spoločenského hrania**. Aby tento výskum teda mal význam, hráči môžu interagovať výhradne iba s dotykovou plochou. Aplikácia si taktiež kladie za cieľ **navýšenie dynamiky**, čo by logicky malo viesť k zrýchleniu celkového procesu hrania. To sa však nepotvrdí v prípade, že samotný dizajn rozhrania nebude dostatočne pochopiteľný a jednoduchý na ovládanie. Používateľské testovanie by sa preto malo sústrediť predovšetkým na pochopenie a uplatňovanie nasledujúcich herných princípov:

1. ovládanie pohybu postavy v móde pre jedného i pre viacerých hráčov
2. orientovanie sa v hernom priestore a porozumenie základným vizualizovaným prvkom (počet zachránených osôb, poškodenie budovy, aktuálne kolo, hráč na ťahu, atď.)
3. vykonávanie akcií (eliminovanie dymu a hasenie ohňa, otváranie a zatváranie dverí, búranie stien, vynášanie zranených osôb, atď.) vrátane porozumenia ich cene
4. odohranie hry bez priveľkých ťažkostí a kritických chybových stavov

Cieľová skupina

Kooperatívna spoločenská hra Záchranári od vydavateľstva Mindok je oficiálne kategorizovaná ako rodinná zábava určená komukoľvek staršiemu ako desať rokov [15]. Reprezentanti vybraní na testovanie by preto v rámci rozmanitosti mali byť ideálne deti vo veku 10 až 16 rokov a dospelí prekračujúci hranicu 25 rokov rozdelení do týchto dvoch skupín:

1. úplní nováčikovia, ktorí o žiadnej verzii hry Záchranári nič nevedia
2. skúsenejší hráči, ktorí hru Záchranári už niekoľkokrát hrali v originálnej podobe

Použitelnosť nie je nevyhnutné testovať výhradne skupinovo a účastníkov nemusí byť priveľa. Dôvod je skrátka ten, že väčšinu najdôležitejších chýb odhalí s najvyššou pravdepodobnosťou už hneď niekoľko prvých jedincov (tie ostatné bývajú najmä subjektívne) [7]. Na základe poskytnutých informácií teda došlo k výberu nasledujúcich uchádzačov:

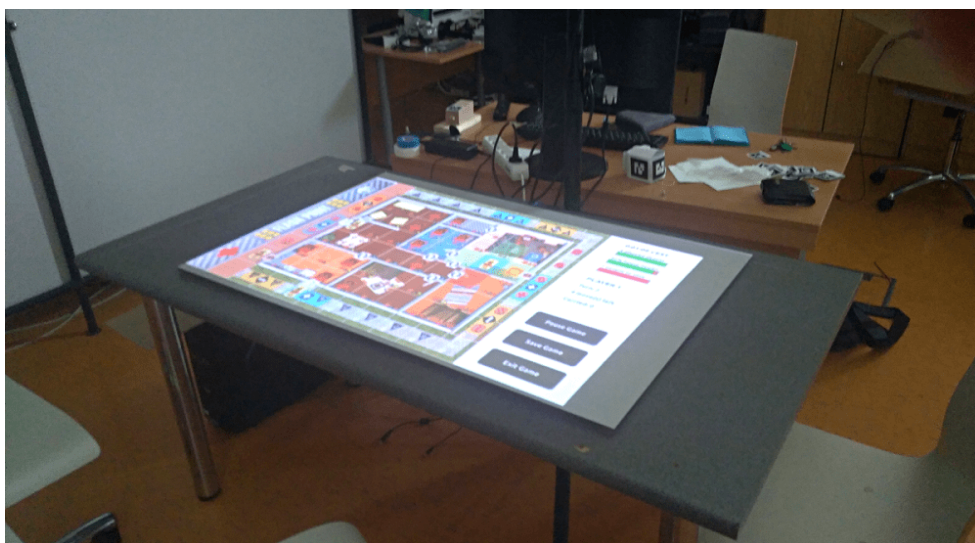
- chlapec mladší ako 10 rokov, ktorý **má** skúsenosti so stolovou hrou Záchranári
- chlapec vo veku 10 až 16 rokov, ktorý **má** skúsenosti so stolovou hrou Záchranári
- muž starší ako 25 rokov, ktorý **má** skúsenosti so stolovou hrou Záchranári
- muž vo veku 15 až 17 rokov, ktorý **nemá** skúsenosti so stolovou hrou Záchranári

Metodológia

Oboznámenie účastníkov testovania s interaktívnym stolom by malo byť obsiahnuté maximálne v troch vetách bez zbytočného rozsiahleho vysvetľovania ovládania. Následne je im už len potrebné zdeliť vyššie definovaný hlavný cieľ hry a stručný popis pravidiel (v prípade, že pôjde o jedinca, ktorý sa s hrou Záchranári ešte nikdy predtým nestretol, malo by mu byť dovolené v rýchlosti si prečítať základnú časť návodu z oficiálneho zdroja [15]).

Role koncových užívateľov, a teda aj samotných hráčov, spĺňajú priamo účastníci testovania. Moderátor (a prípadný potenciálny druhý hráč pri testovaní kooperatívneho hrania s jediným uchádzačom) je zároveň technikom a pozorovateľom zaznamenávajúcim akékoľvek podstatné prejavy prítomných osôb, na základe ktorých dôjde k vyhodnoteniu výsledkov.

Je nevyhnutné, aby používateľské testovanie prebehlo s každou skupinou vzhľadom na časové nároky a dostupnosť technických prostriedkov iba raz, a to v robotickom laboratóriu RoboLab VUT FIT v Brne. Aj na tieto účely je použitý pracovný priestor ARTable3.



Obr. 5.1: Ukážka pripraveného priestoru na používateľské testovanie

Procedúra má nasledujúcu podobu:

1. predstavenie sa a objasnenie cieľa testovania (vrátane ubezpečenia užívateľov, že sa netestujú oni, ale tá aplikácia, takže neexistujú žiadne nesprávne odpovede)
2. stručné oboznámenie účastníkov s hrou Záchranári a jej základnými pravidlami
3. zadanie úlohy testovania prítomným (skupinové odohranie hry na najnižšej obtiažnosti s mapou ľubovoľnej edície v čo najkratšom čase)
4. pozorné sledovanie hráčov zo strany moderátora (so zameraním na študovanie herných prvkov, možné akcie, ovládanie a verbálne prejavy) a zapisovanie priebehu
5. vyplnenie záverečného dotazníka (stačí ústne) s vlastnými návrhmi na zlepšenie (spätnú väzbu si ale moderátor značí i priebežne) a rozlúčka s prípadnou odmenou

Metódy hodnotenia

V priebehu používateľského testovania je pre potreby tejto bakalárskej práce potrebné brať ohľad predovšetkým na nasledujúce kvantitatívne a kvalitatívne metriky:

- miera úspešnosti dokončenia zadanej úlohy s využitím čo najväčšieho počtu možných prostriedkov (kto každý to odohral bez nutnosti pokladania doplňujúcich otázok a zasekávania sa na jednoduchých úkonoch, prípadne na orientácii v GUI)
- čas strávený sústredením sa na grafické užívateľské rozhranie a samotné hranie (predpokladá sa, že ľudia, ktorí hru Záchranári už poznajú, budú rýchlejší)
- správne reagovanie na rôzne prípady a využívanie všetkých druhov akcií (je potrebné sledovať a zaznamenávať, na čo všetko ľudia klikajú, čomu sa naopak vyhýbajú, kedy očividne nerozumejú nejakej udalosti a aké sú ich reakcie na rozličné podnety)
- všetky vyjadrené subjektívne názory (používateľská spokojnosť sa najlepšie vyhodnocuje na základe toho, čo všetko má priebežne a po ukončení hráč na srdci [3])

V rámci testovania je postačujúce vyplniť iba jeden dotazník (je tak možné urobiť len verbálne alebo priamo písomne). S výnimkou prvých dvoch otázok, z ktorých jedna je na štýl **áno/nie**, a priestoru na vyplnenie návrhov zlepšenia aplikácie na druhej strane, vyžaduje predovšetkým odpovede formou zhodnotenia používateľskej spokojnosti na definovanej stupinici. Konkrétna podoba otázok formulára je dostupná v prílohách na konci dokumentu.

5.3 Výsledky používateľského testovania

Používateľské testovanie bolo so všetkými spomínanými jedincami prevedené dňa 24.04.2018 na interaktívnom stole ARTable3 v robotickom laboratóriu RoboLab VUT FIT v Brne. Prvá skupina účastníkov pozostávala z troch ľudí, ktorí so stolovou hrou Záchranári mali v minulosti skúsenosti, pričom druhú okrem samotného moderátora a technika tvoril iba jeden človek bez akýchkoľvek predchádzajúcich súvisiacich znalostí. Výsledky boli podľa plánu zozbierané priebežným zapisovaním prejavov prítomných a prostredníctvom záverečného dotazníka. Táto časť dokumentu sa zaoberá ich vyhodnotením podľa stanovených metód.

Najväčšie chyby aplikácie a možné riešenia

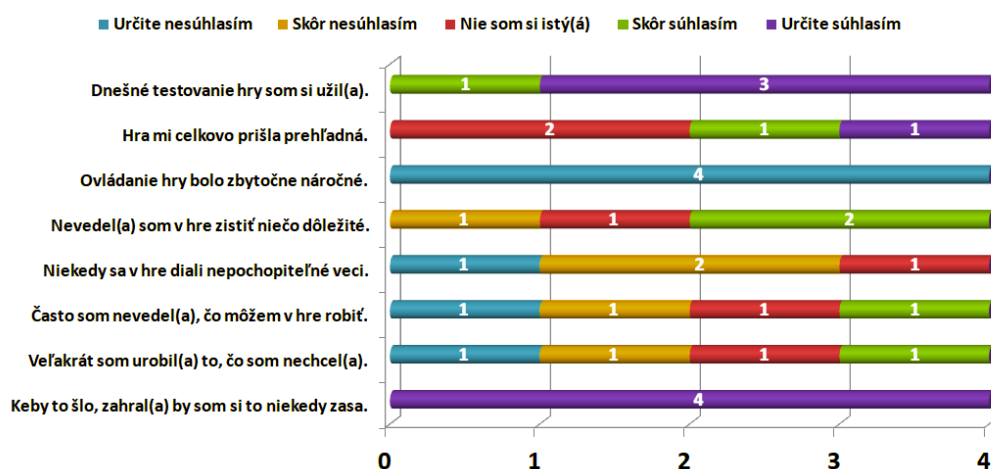
Používateľské testovanie úspešne odhalilo niekoľko väčších chýb týkajúcich sa použiteľnosti programu. Väčšina z nich vychádza priamo z priebežnej verbálnej spätnej väzby účastníkov a ich nechcených problémov, ktorým boli nútení čeliť počas hrania, pričom ich je všetky možné klasifikovať na základe vlastností definovaných v úvode predchádzajúcej sekcie nasledujúcim spôsobom (sú usporiadané od tých zdanlivo najkritickejších po najmenej dôležité):

- **Bezpečnosť:** V oboch skupinách došlo na prevedenie minimálne jednej nechcenej akcie v dôsledku stlačenia nesprávneho prvku používateľského rozhrania, vyústiac dokonca až v predčasné ukončenie kvôli opretiu ruky jedného z prítomných o tlačidlo *Exit Game*. Nežiadaným dotykom interaktívneho stola sa žiaľ nejde vyhnúť úplne, no existujú spôsoby, ako podobným situáciám zabrániť, a to napríklad automatickým ukladaním aktuálnej pozície, vďaka ktorému by hráči pri najhoršom boli schopní vrátiť sa do predchádzajúceho kola. O čosi tradičnejším riešením je taktiež vynucovanie dodatočného potvrdenia závažnejších akcií (vynechanie ťahu a vypnutie hry).

- **Naučiteľnosť a zapamätateľnosť:** V tomto ohľade aplikácia činila problémy predovšetkým jedincovi, ktorý postrádal skúsenosti so stolovou verziou hry Záchranári. Napriek tomu, že mal totiž prístup k návodu s jej pravidlami, nie vždy presne vedel, čo za akcie môže podniknúť a akým spôsobom ich vykonať. V menšej miere to postihlo i druhú skupinu účastníkov, ktorá si napríklad takmer až do konca neuvedomila, že je schopná rozbiť stien budovy. Táto chyba by sa dala dať tomu riešiť priebežným zobrazovaním stručných pomocných inštrukcií počas hry, ktoré by okrem iného bližšie popisovali, čo znamenajú jednotlivé typy vyznačení políčok a žetónov. Vzhľadom na to, že úplne všetky textové prvky aplikácie sú navyše prezentované len v angličtine, by pozitívny vplyv s najvyššou pravdepodobnosťou mala i podpora viacerých jazykov.
- **Účelovosť a efektívnosť:** Testovanie odhalilo akurát negatíva vyplývajúce z niektorých dizajnových rozhodnutí. Jedná sa predovšetkým o umiestnenie úplne všetkých informácií o hre a aktuálnom hráčovi na ľahu mimo hernú mapu, čo však zdanlivo nemá očividné lepšie riešenie, ktoré by sa nepodpísalo na prehľadnosti. Horšie každopádne bolo to, že vybraným účastníkom nepripadalo optimálne rozlišovať medzi interakciou s prvkami nachádzajúcimi sa na podobnom mieste čisto iba presnosťou stlačenia. V aktuálnom prevedení je totiž rozdiel napríklad v tom, či je stlačený priamo žetón dymu (vtedy dochádza na hasenie) alebo políčko, na ktorom sa nachádza (vtedy sa naň hráč akurát presunie). Domnievam sa však, že aj tento problém je riešiteľný zmieňovaným pridaním stručných pomocných inštrukcií, prípadne potrdzovaním akcií, pretože pôvodne navrhovaný ovládací panel (alebo kontextové menu) by istotne uberal jak na celkovej prehľadnosti, tak i na jednoduchosti ovládania.
- **Užitočnosť:** Žiadne konkrétne chyby nájdené neboli, ale existuje nepochybne veľa spôsobov, ako by bolo možné funkcionality aplikácie upraviť a rozšíriť tak, aby to malo prevažne pozitívny vplyv na všetky vlastnosti výsledného programu. Tým sa tento dokument každopádne podrobnejšie venuje až v poslednej časti aktuálnej kapitoly.

Zhodnotenie používateľskej spokojnosti

Používateľskú spokojnosť je možné vyhodnotiť predovšetkým na základe spätnej väzby a odpovedí v zmieňovanom dotazníku, ktorých graf je k dispozícii nižšie (pozri obrázok 5.2).



Obr. 5.2: Graf odpovedí poskytnutých v rámci používateľského testovania

Podľa kategorizácie na metriky definované v predchádzajúcej sekcii je v prvom rade vhodné diskutovať o takzvanej **miere úspešnosti** postupovania v hre a **využívania jednotlivých akcií**. Ani jeden účastník nepovažoval samotné ovládanie hry za náročné. Orientácia v tom, čo všetko a akým spôsobom je v ktorej situácii možné vykonať, však už bola podstatne nerovnomerná. Každý jeden absolvent testovania totiž vyjadril o čosi rozdielny názor na súčasnú interpretáciu možných akcií a minimálne raz vykonal nejakú nechcenú, z čoho logicky vyplývajú aj niektoré z vyššie spomínaných väčších chýb aplikácie.

Čas strávený sústredením sa na rozhranie bol podľa očakávaní kratší v skupine, ktorá už mala skúsenosti so stolovou hrou Záchranári. Iba polovica účastníkov ale tvrdila, že im aplikácia prišla dostatočne prehľadná, pričom tá druhá mala vzhľadom na určité výhrady neutrálny názor. Každý prítomný totiž prejavil problém zistiť aspoň nejakú informáciu bez toho, aby sa voľakto opýtal, no nikto až na človeka, ktorý nebol dostatočne oboznámený s pravidlami hry, nevyjadril nepochopenie udalostí plynúcich z odohraných akcií.

Navzdory všetkým problémom v oboch skupinách panovala uvoľnená atmosféra a každý z účastníkov dal najavo, že jeho celková prvá skúsenosť s testovaným programom nebola negatívna. Všetci tvrdili, že by si v budúcnosti výslednú hru určite radi znova zahráli, a to dokonca i jedinec, ktorému jej stolová verzia nič nehovorila. Aplikácia má preto nepochybne potenciál dosiahnuť značne vysokú mieru pozitívnej používateľskej spokojnosti.

Návrhy ďalších vylepšení a rozšírení programu

Okrem vyššie zmieňovaných riešení najväčších odhalených problémov je zo spätnej väzby používateľov možné vyvodiť aj niekoľko ďalších potenciálnych vylepšení aplikácie. Jedná sa v prvom rade o **pridanie viacerých multimediálnych prvkov** (predovšetkým animácií, prepracovanejšej živšej grafiky a zvukov), na ktoré sa vzhľadom na obmedzené prostriedky a časové nároky nebolo možné sústrediť už v aktuálnej verzii hry. Za zmienku taktiež stojí návrh **doplnenia špeciálnych herných módov**, a to konkrétne napríklad týchto:

- **Režim hry pre pokročilých:** Tento mód zavádza nové pravidlá a elementy (vozidlá, rôzne špecializácie záchranárov, nebezpečné látky, rozžhavenie, atď.), ktoré sú podrobne špecifikované na oficiálnej stránke hry (v návode) [15].
- **Extrémny režim hry:** V tomto móde sú všetky časti budovy (vyčlenené na základe stien a dverí), v ktorých sa nenachádza žiadny záchranár, kompletne zakryté, aby ich hráči neboli schopní vidieť skôr, než do nich reálne dostanú svoje postavy.
- **Dynamický režim hry:** Tento mód urýchľuje šírenie ohňa a umožňuje všetkým hráčom na základe informácií o ich skutočných pozíciách získaných zo záznamu hĺbkovej kamery realizovať svoje akcie naraz počas každého jedného kola.

Na záver by v rámci zlepšenia **interaktivity aplikácie** bolo možné urobiť objekty reprezentujúce hráčov transparentnými a viazať ich na pozície skutočných figúrok. Pohyb záchranárov by teda prebiehal presúvaním reálnych fyzických predmetov (snímaných hĺbkovou kamerou), pričom všetky ostatné akcie by naďalej spočívali v stláčaní vybraných herných elementov na dotykovej ploche. Súčasný dizajn používateľského rozhrania je zámerne koncipovaný tak, aby táto varianta nevyžadovala priveľa úprav v prevedenej implementácii.

Kapitola 6

Záver

Cieľom mojej bakalárskej práce bolo realizovať vybranú stolovú hru s využitím interaktívneho stola za účelom poskytnutia pozitívnej používateľskej skúsenosti bez straty najvýznamnejších aspektov definujúcich jej originál. Na ilustráciu s prevedením postupu dosiahnutia tohto zámeru bola zvolená kooperatívna hra Záchranári od Mindok a pracovný priestor s názvom ARTable3, dostupný v robotickom laboratóriu VUT FIT v Brne. Kľúčom k úspechu bolo naštudovanie informácií týkajúcich sa predovšetkým používateľských rozhraní, rozšírenej reality a spomínaného interaktívneho zariadenia, pričom jedným z najpodstatnejších výstupov práce je práve postup prevodu stolovej hry do počítačovej podoby so zachovaním nielen sociálnych aspektov, ale i hratelnosti pri odstránení fyzických objektov originálu.

Počínajúc nadobudnutím všetkých potrebných teoretických poznatkov som sa zamerával v prvom rade na analýzu konkrétnych potrieb interakcie a požiadaviek kladených na program. Pokračoval som vytvorením návrhu modernizácie hry a naplánovaním technického výskumu jej kvality. Následne som sa venoval implementácii počítačovej verzie funkčného prototypu, sprevádzanej pridaním snímania vstupov z dotykovej plochy interaktívneho stola a premietania výstupného obrazu na jeho povrch v závislosti na spracovaní získaných podnetov. Posledným krokom bola realizácia testovania na skutočných používateľoch a vyhodnotenie spätnej väzby, na základe ktorej som poskytol i nejaké možnosti vylepšenia aplikácie.

Ďalším výsledkom mojej práce je samotný vzniknutý program, ktorý zachováva kľúčové sociálne aspekty spoločenskej zábavy a vyzdvihuje predovšetkým dynamiku hrania automatizovaním všetkých najnudnejších procesov priebežných príprav hernej mapy. Vzájomná komunikácia medzi hardvérovými komponentmi je v ňom riešená uplatnením niektorých princípov robotického operačného systému prostredníctvom vybraných balíkov pracovného prostredia s prekladovým systémom catkin. Bol riadne otestovaný so zameraním na funkčnosť a použiteľnosť, pričom zo spracovanej spätnej väzby účastníkov výskumu kvality aplikácie je možné vyvodiť, že v hráčoch v značnej miere dosahuje používateľskú spokojnosť.

Hra má napriek tomu nepochybne priestor na vylepšenia, a to v prvom rade vyriešením nedostatkov plynúcich z vybraných názorov jedincov prítomných na jej používateľskom testovaní. V rámci efektívnejšieho vtiahnutia hráčov do diania by bolo dobré pridať aj viac multimediálnych prvkov a špeciálnych módov s rozšírením funkcionality programu. Rozhranie aplikácie navyše umožňuje doplnením implementácie kalibrácie hĺbkovej kamery interaktívneho stola a spracovania jej obrazu s minimálnymi úpravami zdrojového kódu reprezentovať záchranárov skutočnými figúrkami. Aktuálna verzia programu každopádne najkľúčovejšie súvisiace požiadavky spĺňa, čo z nej činí vhodný základ pre všetky navrhované vylepšenia.

Literatúra

- [1] Baecker, R. M.: *Readings in Human-Computer Interaction: Toward the Year 2000 (Interactive Technologies)*. Morgan Kaufmann Publishers, 1995, ISBN 978-1558602465.
- [2] Bimber, O.; Raskar, R.: *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A K Peters, 2005, ISBN 978-1568812304.
- [3] Dumas, J. S.; Redish, J. C.: *A Practical Guide to Usability Testing*. Intellect, 1999, ISBN 978-1841500201.
- [4] Geroimenko, V.: *Augmented Reality Art: From an Emerging Technology to a Novel Creative Medium*. Springer, 2014, ISBN 978-3-319-06202-0.
- [5] Gobet, F.; Retschitzki, J.; Voogt, A.: *Moves in Mind The Psychology of Board Games*. Psychology Press, 2004, ISBN 978-1135425135.
- [6] Gutwin, C.; Greenberg, S.: *The Importance of Awareness for Team Cognition in Distributed Collaboration. Team Cognition: Understanding the Factors That Drive Process and Performance*, 2004: str. 201.
- [7] Krug, S.: *Nenutte uživatele přemýšlet!* Computer Press, 2007, ISBN 80-251-1291-8.
- [8] Marinilli, M.: *The Theory Behind User Interface Design, Part One*. [online; navštívené 23.04.2018].
URL <https://www.developer.com/design/article.php/1545991>
- [9] Myers, G. J.: *The Art of Software Testing, Second Edition*. Wiley, 2004, ISBN 978-0471469124.
- [10] Open Source Robotics Foundation: *ros.org*. [online; navštívené 26.04.2018].
URL <http://wiki.ros.org/>
- [11] Riverbank Computing Limited: *PyQt5 Reference Guide*. [online; navštívené 29.04.2018].
URL <http://pyqt.sourceforge.net/Docs/PyQt5/>
- [12] Shaer, O.; Hornecker, E.: *Tangible User Interfaces: Past, Present, and Future Directions. Foundations and Trends in Human-Computer Interaction*, ročník 3, č. 1–2, 2009: str. 1–137.
- [13] Shneiderman, B.; Plaisant, C.: *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley Computing, 2009, ISBN 0-321-53735-1.

- [14] Sieber, R.; Wiesmann, S.; Schnabel, O.: *1. Graphical User Interface - Layout and Design*. [online; navštívené 23.04.2018].
URL http://www.e-cartouche.ch/content_reg/cartouche/ui_access/en/
- [15] SoftGate Systems: *MINDOK - moderní společenské hry*. [online; navštívené 21.04.2018].
URL <http://www.mindok.cz/>
- [16] The Qt Company Ltd.: *QT Documentation*. [online; navštívené 16.04.2018].
URL <http://doc.qt.io/>
- [17] Wikimedia Foundation: *Testování softwaru*. [online; navštívené 15.04.2018].
URL https://cs.wikipedia.org/wiki/Testov%C3%A1n%C3%AD_softwaru
- [18] Wikimedia Foundation: *User interface*. [online; navštívené 17.04.2018].
URL https://en.wikipedia.org/wiki/User_interface

Príloha A

Obsah CD

Na priloženom pamäťovom médiu je možné nájsť nasledujúce doplňujúce materiály:

- **Plagát:** súbor *poster.pdf*
- **Demonštračné video:** súbor *video.mp4*
- **Technická správa:** súbor *xbacov04.pdf*
- **Návod k hre Záchranári:** súbor *manual.pdf*
- **Zdrojové súbory programu:** adresár *xbacov04*
- **Zdrojové súbory technickej správy:** adresár *latex*

Príloha B

Dotazník

DOTAZNÍK SPOKOJNOSTI

Vyplňte prosím tento dotazník. Do správnych políček umiestnite krížik.

1. V akej vekovej kategórii sa nachádzate?

☐

menej ako 10 rokov

☐

10 až 16 rokov

☐

17 až 25 rokov

☐

viac ako 25 rokov

2. Hrali ste už v minulosti spoločenskú hru Záchranári?

☐

áno

☐

nie

3. Označte na každom riadku políčko, ktoré najviac vystihuje Váš názor.

	Určite nesúhlasím	Skôr nesúhlasím	Nie som si istý(á)	Skôr súhlasím	Určite súhlasím
Dnešné testovanie hry som si užil(a).					
Hra mi celkovo prišla prehľadná.					
Ovládanie hry bolo zbytočne náročné.					
Nevedel(a) som v hre zistiť niečo dôležité.					
Niekedy sa v hre diali nepochopiteľné veci.					
Často som nevedel(a), čo môžem v hre robiť.					
Veľakrát som urobil(a) to, čo som nechcel(a).					
Keby to šlo, zahral(a) by som si to niekedy zasa.					

4. Napíšte kamkoľvek na druhú stranu (stačí cez odrážky), aké zmeny by ste pre hru Záchranári na interaktívnom stole do budúcnosti najviac uvítali.

Príloha C

Odpovede na dotazník

Táto tabuľka reprezentuje počet používateľov vyjadrujúcich konkrétne názory na dané tvrdenia dotazníka spokojnosti, ktorý bol každým účastníkom testovania aplikácie vyplnený:

	Určite nesúhlasím	Skôr nesúhlasím	Nie som si istý(á)	Skôr súhlasím	Určite súhlasím
Dnešné testovanie hry som si užil(a).	0	0	0	1	3
Hra mi celkovo prišla prehľadná.	0	0	2	1	1
Ovládanie hry bolo zbytočne náročné.	4	0	0	0	0
Nevedel(a) som v hre zistiť niečo dôležité.	0	1	1	2	0
Niekedy sa v hre diali nepochopiteľné veci.	1	2	1	0	0
Často som nevedel(a), čo môžem v hre robiť.	1	1	1	1	0
Veľakrát som urobil(a) to, čo som nechcel(a).	1	1	1	1	0
Keby to šlo, zahral(a) by som si to niekedy zasa.	0	0	0	0	4

Príloha D

Používateľský manuál

Hra je obsiahnutá v balíku *robotického operačného systému* (*Robot Operating System, ROS*) s pomenovaním **xbacov04**, ktorý pozostáva predovšetkým zo zdrojového kódu umiestneného v adresári **src**, textového dokumentu zohrávajúceho úlohu vstupov pre preklad s názvom **CMakeLists**, *XML* súboru **package** s metainformáciami a v neposlednom rade inštaláčného skriptu napísaného v jazyku *Python*, **setup**. Zdrojové súbory je po zakomentovaní vybraných riadkov v **rescuers_gui.py** možné preložiť aj ako desktopovú aplikáciu ovládateľnú myšou s využitím prostredia *QT Creator* s multiplatformným rámcom *Qt 5* (na akomkoľvek operačnom systéme, kde je spomínané vývojové prostredie možné nainštalovať), no na jej spozajzdnenie pre interaktívny stôl je nevyhnutný počítač s operačným systémom podporujúcim nielen *Qt 5* (s rozširujúcimi modulmi *PyQt5*), ale i *CMake* a samotný *ROS*.

Inštalácia balíka

Za predpokladu, že sú na cieľovom zariadení nainštalované všetky vyššie zmieňované prerekvizity, je akurát potrebné mať vytvorené *pracovné prostredie* (*workspace*) prekladového systému *catkin*, do ktorého musí byť balík **xbacov04** presunutý. Po spustení *roscore* cez terminál stačí zmeniť aktuálny priečinok na spomínané pracovné prostredie a zadať príkaz "*catkin_make*" (keby to náhodou nestačilo, odporúča sa vyskúšať ešte aj "*catkin_make install*"), po ktorého úspešnom prevedení je inštaláciu možné považovať za ukončenú.

Spustenie balíka

Na to, aby bolo možné hru spozajzdniť, je nevyhnutné okrem vyššie zmieňovaných prerekvizít mať samozrejme zapojené (do zariadenia, cez ktoré je aplikácia spúšťaná) a zapnuté zvyšné komponenty interaktívneho stola (projektor, dotyková plocha, hĺbková kamera). Na počítači nesmú chýbať ani ich ovládače a podružné balíky umiestnené v spomínanom *pracovnom prostredí* (*workspace*) prekladového systému *catkin* (mali by byť dostupné v *GitHub* repozitári na adrese <https://github.com/robofit/artable>). Taktiež je potrebné, aby súbor **rescuers_gui.py** mal prístupové právo *execute* (keby tomu tak nebolo, je to možné zmeniť príkazom "*chmod +x <cesta_k_súboru/rescuers_gui.py>*"). Za predpokladu, že sú všetky tieto podmienky splnené, stačí v zachovanom poradí postupovať nasledujúcim spôsobom:

1. **Zadanie príkazu "*roscore*" do okna terminálu:** Tento krok slúži na spustenie kolekcie uzlov a programov potrebných pre beh "*robotického operačného systému*". Pokiaľ už *roscore* predom spustený je, znova to samozrejme vykonávať netreba.

2. Rozmiestnenie štyroch hlavných značiek (*markerov*) na dotykovú plochu a zadanie príkazu "*roslaunch art_bringup bringup.launch*" do ďalšieho okna terminálu: Tento krok zabezpečí správne spustenie ovládačov a uzlov komponentov interaktívneho stola. Zmieňované značky je vhodné rozložiť podľa obrázka D.1.

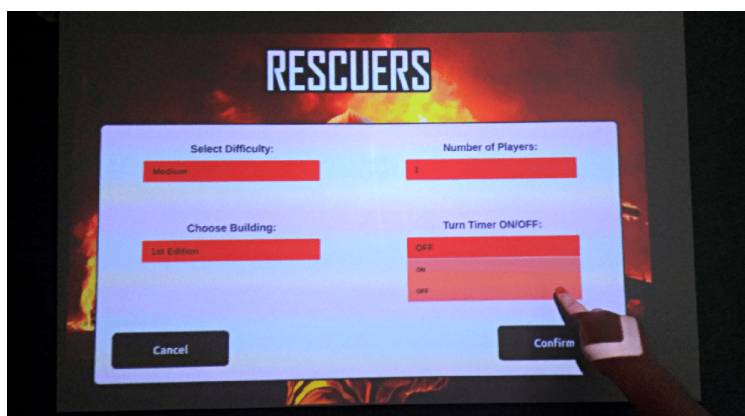


Obr. D.1: Ukážka rozmiestnenia značiek na interaktívnom stole

3. Zadanie príkazu "*roslaunch xbacov04 rescuers_gui.py*" do ďalšieho okna terminálu: V prípade, že sa jedná o prvé otváranie hry od spustenia *roscore*, prevedenie tohto príkazu nejakú chvíľu potrvá a výsledkom bude zobrazenie bieleho bodu na čiernej scéne premietanej na dotykovú plochu, ktorý je potrebné stlačiť. To treba urobiť pre každý jeden z nich až do doby, než sa konečne ukáže hlavná obrazovka aplikácie.

Ovládanie hry

Úplne všetky akcie na úvodnej i hernej obrazovke sú v aktuálnej verzii aplikácie uskutočňované stlačením konkrétnych prvkov rozhrania, a to buď na dotykovej ploche alebo kliknutím myšou (pozri obrázok D.2). O bližšom popise každého jedného z nich podrobne informuje **technická správa**, pričom pravidlá hry sú okrem nej dostupné aj na internetovej stránke, na ktorú sa dá dostať prostredníctvom tlačidla *Instructions* (listovanie v nej a návrat naspäť do hlavného okna programu je však možné uskutočniť jedine priamo na počítači myšou).



Obr. D.2: Ukážka menenia nastavení hry na úvodnej obrazovke